Proceedings of DETC'02:
ASME 2002 DESIGN ENGINEERING TECHNICAL CONFERENCES AND COMPUTERS AND INFORMATION IN ENGINEERING CONFERENCE
Montreal, Canada
September 29–October 2, 2002

DETC2002/MECH-34325

PLANNING FOR KINEMATICALLY SMOOTH MANIPULATOR TRAJECTORIES

Zan Mi
Center for Computer Aided Design
The Department of Mechanical and Industrial Engineering
THE UNIVERSITY OF IOWA

Jingzhou Yang
Center for Computer Aided Design
The Department of Mechanical and Industrial Engineering
THE UNIVERSITY OF IOWA

Karim Abdel-Malek
Center for Computer Aided Design
The Department of Mechanical and Industrial Engineering
THE UNIVERSITY OF IOWA
Iowa City, IA 52242
Tel. 319-335-5676
Fax. 319-335-5669
amalek@engineering.uiowa.edu

Laurent Jay
The Department of Mathematics
THE UNIVERSITY OF IOWA
Iowa City, IA 52242-1419
ljay@math.uiowa.edu

ABSTRACT

This paper presents a general methodology and accompanying formulation for planning kinematically smooth path trajectories for serial robot manipulators. Starting from an initial point on the path, it is required to traverse a path trajectory without halting the motion (typically due to switch from one inverse solution to another). The problem focuses on determining a starting configuration at the initial point on the path. The problem is formulated in terms of a constraint function and characterized by a system of differential-algebraic equations (DAEs) of index 2. These DAEs are solved numerically by Runge-Kutta (RK) methods while a best starting configuration is chosen using a cost-function driven optimization method. The formulation is demonstrated through a planar 3 degree of freedom robot manipulator.

Keywords: kinematically smooth, trajectories, path, planning, differential-algebraic equations

INTRODUCTION

In many robotic applications, it is important to plan smooth path trajectories that would enable the end-effector to complete the motion uninterrupted. We define a *kinematically smooth trajectory* as one that does not admit changes in inverse kinematic solutions during its motion. For example, in robotic welding applications, it is imperative that the end-effector carrying the welding fixture be allowed to complete the path without a discontinuity in the path, and without halting of the robot's motion. Similarly, in surgical robotic applications, in simulating path trajectories for human motion, or in planning interference free paths for cooperative robots, it is important to rigorously identify which starting configuration will yield a kinematically smooth path, one that would not be interrupted during its motion.

The problem of determining kinematically smooth path trajectories has not really been addressed in the literature. However, delineation of singular behavior where the manipulator may or may not be able to cross was addressed by many researchers and is typically based on a null-space criterion of the manipulator's Jacobian [1-8].

Specific crossability criteria through barriers in the workspace were reported by Nielsen *et al.* [9]. The fundamental concept of crossable and noncrossable surfaces inside a manipulator's workspace was addressed by Oblak and Kohli [10]. Although their report has touched upon a crossable

surface using a singularity criterion of the Jacobian, the paper does not present a unified method for identifying control problems neither does it determine all barriers inside the workspace in closed form. A criterion to define possible motion (so-called feasible trajectory) from a singularity was presented by Chevallereau and Daya [11] and Chevallereau [12].

Haug *et al.* [13] presented a numerical algorithm for identifying and analyzing barriers to output control of manipulators using first- and second-order Taylor approximations of the output in selected directions. Haug and colleagues showed that the output velocity in the direction normal to such curves and surfaces must be zero [14] and manipulator boundaries were consequently mapped.

The method presented in this paper is based on recent results by this group where singular surfaces in manipulator workspaces were delineated and acceleration-based crossability criteria were defined [15-19].

PROBLEM DEFINITION

The objective of this work is to identify starting configurations of the robotic arm in such a way to allow for a kinematically smooth path trajectory. We will assume that we are given the complete characteristics of the robot manipulator, its joint limits, and a desired path trajectory.

From the Denavit-Hartenberg representation method, we obtain the position vector of a point on the end-effector as

$$\mathbf{p}(t) = \mathbf{F}(\mathbf{q}(t)) \tag{1}$$

where $\mathbf{p}$ represents the coordinates of a path, $\mathbf{F}$ stands for the position coordinates of the end-effector of a kinematic structure and $\mathbf{q}$ is the vector of joint variables. Differentiating with respect to time, we obtain

$$\dot{\mathbf{p}} = \mathbf{F}_{\mathbf{q}}\dot{\mathbf{q}} \tag{2}$$

where $\mathbf{F}_{\mathbf{q}}$ is the (3 x *n*) Jacobian of $\mathbf{F}$ with respect to $\mathbf{q}$, where *n* is the number of degrees of freedom (DOF) of the robot.

Our non-crossable surfaces here mean within the structure's workspace those surfaces that can not be crossed in at least one direction, though they may be crossed in some particular direction. For the points on those non-crossable surfaces, a structure can have many different configurations with end-effector right on the surface points. The structure will not be able to cross the surface only when it has the same configuration as that of the non-crossable surface.

Now it is possible to better define the notion of delineating a kinematically smooth trajectory. It is to determine an appropriate initial configuration that would enable the end-effector to continuously move from this configuration along the path, uninterrupted. We also define an interruption as a singular configuration on a singular surface resulting from the intersection of the path trajectory with a control barrier, and where this surface is non-crossable. If those conditions are met, the trajectory is kinematically smooth (i.e., the end effector can move smoothly across a non-crossable barrier).

PROBLEM FORMULATION

We first develop the constraint equations necessary to ensure that the inverse kinematic solution will constrain the end-effector to remain on the path. From Eqs. (1) and (2), we implement the pseudoinverse $\mathbf{F}_{\mathbf{q}}^{\#}$ [20] such that

$$\dot{\mathbf{q}} = \mathbf{F}_{\mathbf{q}}^{\#}\dot{\mathbf{p}} \tag{3}$$

where $$\mathbf{F}_{\mathbf{q}}^{\#} = \mathbf{F}_{\mathbf{q}}^{T}(\mathbf{F}_{\mathbf{q}}\mathbf{F}_{\mathbf{q}}^{T})^{-1} \tag{4}$$

Since $\mathbf{F}_{\mathbf{q}}$ is an $m \times n$ matrix, where $m < n$, and suppose $\mathbf{q}$ is not a singular configuration, then Rank $(\mathbf{F}_{\mathbf{q}}) = m$ and thus we have

$$\dot{\mathbf{q}} = \mathbf{F}_{\mathbf{q}}^{T}(\mathbf{F}_{\mathbf{q}}\mathbf{F}_{\mathbf{q}}^{T})^{-1}\dot{\mathbf{p}} \tag{5}$$

Introduce a vector variable $\mathbf{z}$, the above equation can be converted to the equivalent system of equations

$$\dot{\mathbf{q}} = \mathbf{F}_{\mathbf{q}}^{T}(\mathbf{q})\mathbf{z} = \mathbf{f}(\mathbf{q}, \mathbf{z}) \tag{6.a}$$

$$\mathbf{0} = \mathbf{F}(\mathbf{q}) - \mathbf{p}(t) = \mathbf{g}(\mathbf{q}, t) \tag{6.b}$$

Differentiate (6.b) with respect to t and substitute (6.a) into $\dot{\mathbf{q}}$, we have

$$\mathbf{g}_t(\mathbf{q}, t) = \mathbf{F}_{\mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{p}}(t) = \mathbf{F}_{\mathbf{q}}(\mathbf{q})\mathbf{F}_{\mathbf{q}}^{T}(\mathbf{q})\mathbf{z} - \dot{\mathbf{p}}(t) = \mathbf{0} \tag{7}$$

Equation (7) is written as a constraint to be satisfied while planning the motion. The above equations are indeed differential-algebraic equations (DAEs) of index 2 and can be addressed using Runge-Kutta methods.

RUNGE-KUTTA METHODS FOR INDEX 2 DAES

We briefly present the numerical method considered in our numerical experiments to approximate a solution to the system of DAEs (6). Consider more generally the semi-explicit and non-autonomous system of index 2 DAEs

$$y' = f(t, y, z), \qquad 0 = g(t, y) \tag{8}$$

where f and g are supposed to be sufficiently differentiable and $g_y(t, y) f_z(t, y, z)$ is assumed to be invertible. The initial conditions t_0, y_0, z_0 are supposed to be consistent with Eq. (8), i.e.,

$$0 = g(t_0, y_0), \qquad 0 = g_y(t_0, y_0) f(t_0, y_0, z_0) + g_t(t_0, y_0).$$

The application of Runge-Kutta (RK) methods to the above system of DAEs (8) can be expressed as follows

$$y_{n+1} = y_n + h_n \sum_{j=1}^{s} b_j f(t_n + c_j h_n, Y_{nj}, Z_{nj}) \tag{9}$$

where the internal stages Y_{ni}, Z_{ni} for $i = 1, \ldots, s$ are solution of

$$Y_{ni} = y_n + h_n \sum_{j=1}^{s} a_{ij} f(t_n + c_j h_n, Y_{nj}, Z_{nj}), \qquad 0 = g(t_n + c_i h_n, Y_{ni}).$$

RK methods are characterized by the set of RK coefficients $(a_{ij})_{i,j=1}^{s}$ and they are based on a quadrature formula given by coefficients $(b_j, c_j)_{j=1}^{s}$. Stiffly accurate RK methods by definition satisfy $a_{sj} = b_j$ for $j = 1, \ldots, s$. They are of particular interest since y_{n+1} then automatically satisfies $0 = g(t_{n+1}, y_{n+1})$ where $t_{n+1} = t_n + h_n$. For such RK methods, it is also natural to take $z_{n+1} := Z_{ns}$. For RK methods whose coefficients satisfy $a_{1j} = 0$ for $j = 1, \ldots, s$, it is natural to take

$Z_{n1} := z_n$. Other alternatives based on RK coefficients are half-explicit RK methods [21-23], partitioned RK methods [24,25], and SPARK methods [26]. The RK method used in our numerical experiments is the 2-stage Lobatto IIIA method, simply speaking the trapezoidal rule. Convergence of order 2 for the y-component is achieved, meaning that the global error of the y-component on a finite interval between the exact solution and the trapezoidal rule approximation is bounded by $Const \cdot h^2$ where $h := \max_n h_n$. Detailed convergence results for Lobatto IIIA methods can be found in Hairer *et al*. [22], Hairer and Wanner [23] and Jay [27]. The Butcher tableau coefficients

c_i	a_{ij}
	b_j

of the 2-stage Lobatto IIIA RK method is given as follows

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

For this method we obtain the following system of nonlinear equations for y_{n+1} and z_{n+1}

$$\mathbf{0} = \mathbf{y}_{n+1} - \left(\mathbf{y}_n + \frac{h_n}{2} \left(\mathbf{f}(t_n, \mathbf{y}_n, \mathbf{z}_n) + \mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}, \mathbf{z}_{n+1}) \right) \right) \quad (10)$$

$$\mathbf{0} = \mathbf{g}(t_{n+1}, \mathbf{y}_{n+1}).$$

These nonlinear equations can be solved iteratively by modified Newton iterations with approximate Jacobian matrix

$$\boldsymbol{J} := \begin{bmatrix} \boldsymbol{I} & -\frac{h_n}{2} \boldsymbol{f}_z(t_n, \mathbf{y}_n, \mathbf{z}_n) \\ \boldsymbol{g}_y(t_n, \mathbf{y}_n) & \boldsymbol{O} \end{bmatrix}. \quad (11)$$

OPTIMIZATION

Normally, given a desired position of a redundant system, inverse kinematics is implemented to determine an infinite number of configurations that yield the correct answer, and one of these solutions is typically selected. Given a path and the barrier information of the system, we first use the method described above to transform the problem into a DAE system of equations, then use Runge-Kutta method to find an initial configuration that will admit a kinematically smooth path. However due to the redundancy property of the system, there will still be many feasible initial configurations. How to choose the best initial configuration is indeed an optimization problem. Our purpose becomes to find a best start configuration, not only to admit kinematically smooth motion, but also to minimize a cost function during the movement. The concept of optimizing a given cost function towards calculating an appropriate solution in a computationally effective manner was addressed by Abdel-Malek *et al*. [19].

To simplify the optimization process, some points on the path are chosen, whereat the cost function is calculated. Within a number of feasible initial configurations, which will admit kinematically smooth motion, the solution with the minimum cost is readily identified.

EXAMPLE

Consider a planar 3-DOF manipulator arm shown in Fig. 1, which is comprised of three links and three revolute joints. Inequality constraints are imposed on the three joints as follows:

$$-\frac{p}{3} \le q_i \le \frac{p}{3}, \quad i = 1, 2, 3$$

We will follow the position of a point on the end-effector with the following position vector

$$\mathbf{F}(\mathbf{q}) = \begin{bmatrix} 4\cos(q_1) + 2\cos(q_1 + q_2) + \cos(q_1 + q_2 + q_3) \\ 4\sin(q_1) + 2\sin(q_1 + q_2) + \sin(q_1 + q_2 + q_3) \end{bmatrix} \quad (12)$$

where $\mathbf{q} = [q_1, q_2, q_3]^T$. The Jacobian matrix $\mathbf{F}_q$ is

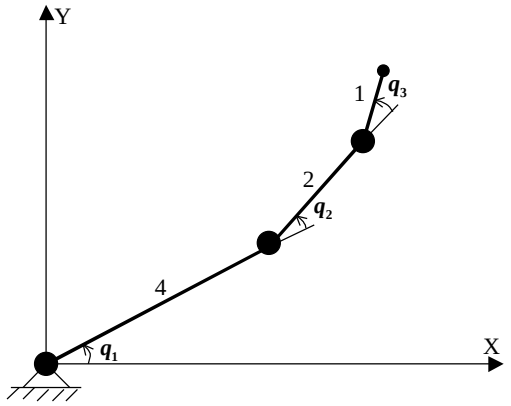

Figure 1. A planar 3-DOF manipulator

$$\mathbf{F}_q(\mathbf{q}) = \begin{bmatrix} -4\sin(q_1) - 2\sin(q_1 + q_2) - \sin(q_1 + q_2 + q_3) & \\ 4\cos(q_1) + 2\cos(q_1 + q_2) + \cos(q_1 + q_2 + q_3) & \\ -2\sin(q_1 + q_2) - \sin(q_1 + q_2 + q_3) & -\sin(q_1 + q_2 + q_3) \\ 2\cos(q_1 + q_2) + \cos(q_1 + q_2 + q_3) & \cos(q_1 + q_2 + q_3) \end{bmatrix} \quad (13)$$

As first identified by Yeh [15], there are three types of singular behavior used to delineate the barriers to motion of a robotic manipulator: (1) Jacobian singularities; (2) Joint limit singularities; (3) Coupled singular behavior. The singularity sets for this example were obtained by Yeh [15] and are shown in Fig. 2. From these sets, non-crossable surfaces (curves in this case) are identified [16,19] and one of them is shown in Fig. 3.

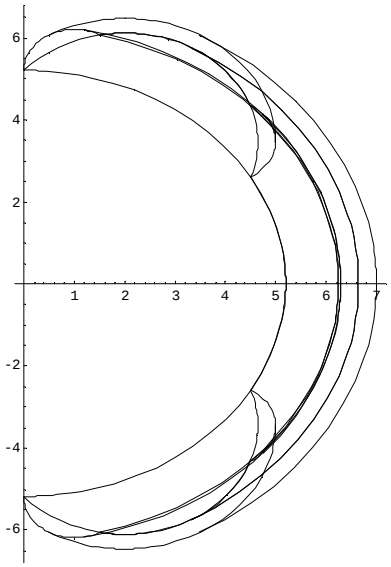

Figure 2. Singular curves

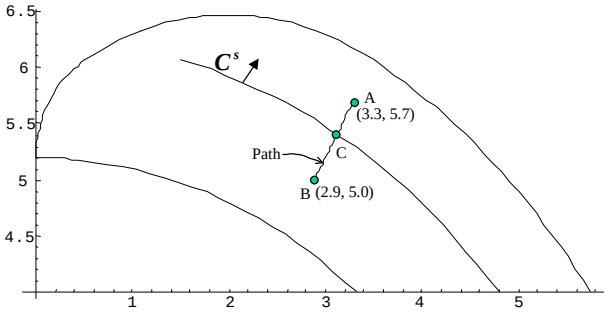

Figure 3. A non-crossable singular curve and a path

Indeed, singular curves can be divided into crossable and non-crossable curves. Crossable curves indicate that the end-effector with this singular configuration can cross the curve, i.e., no need to go back to change start configuration and return. On the contrary, when the end-effector gets to the non-crossable curve with the specific singular configuration, the motion will be interrupted and the system will have to switch to another configuration and attempt the crossing again. Fig. 3 shows a non-crossable curve and a path, which the end-effector must follow. As shown by the arrow, the non-crossable curve $\boldsymbol{C}^s$ can only be crossed in one direction. Therefore, when the manipulator starts from point A following the path to point B, it will not be able to cross this curve if it gets to the singular configuration at the intersection point C.

To calculate a good initial configuration at point A, the following procedure is implemented. More generally, for any given path trajectory, the path must first be evaluated to determine if the path intersects with any non-crossable singular curve. If such an intersection is determined, the non-crossable singular configuration at the intersection point will be found. For the 3-DOF example, the path trajectory is defined by

$$\mathbf{p}(t) = \begin{bmatrix} t \\ \sqrt{3}t \end{bmatrix} \tag{14}$$

The coordinates of the initial point A and end point B on the path are specified as (3.3, 5.7) and (2.9, 5.0), respectively. The path intersects with a non-crossable curve

$$\boldsymbol{C}^s:\ q_2 = p/3,\ q_3 = -p/3,\ -p/3 \le q_1 \le p/3$$

The singular configuration at the intersection point C is found to be $\mathbf{q}^s = [\mathbf{0.7662}, p/3, -p/3]^T$.

At the initial point, the optimization algorithm is used to find an inverse kinematics solution for the initial configuration $\mathbf{q_0}$. Subsequently, Runge-Kutta method for DAE of index 2 described above is applied to do the constrained integration to calculate the configuration $\mathbf{q}^c$ of the manipulator at point C. Now this configuration is compared with the singular configuration $\mathbf{q}^s$ to check if

$$\left\|\mathbf{q}^c - \mathbf{q}^s\right\|^2 < e \tag{15}$$

where e is a small positive number. If the inequlity (15) is true, we then return to the initial point A to calculate a different initial configuration. We repeat the procedure until a satisfying initial configuration is calculated. During the integration, if any configuration on the path violates the joint limit constraints, the algorithm is interrupted immediately and the loop is repeated to re-integrate.

At the initial point A, $t = 3.3$, $\dot{\mathbf{p}} = [1,\ \sqrt{3}]^T$, $\mathbf{q_0}$ is found to be $[0.8030,\ 0.7816,\ -0.5946]^T$, substitute $\mathbf{q_0}$ into Eq. (7) to get $\mathbf{z_0} = [1.1276,\ 2.1970]^T$. Step length h is selected to be –0.0005. At the following step, we calculate $t_1 = t_0 + h = 3.3 - 0.0005 = 3.2995$, and the results of the integration yields $\mathbf{q_1} = \mathbf{Q_2} = [0.8036,\ 0.7803,\ -0.5945]^T$ and $\mathbf{Z_2} = [-3.9613,\ -7.7424]^T$. We can make $\mathbf{z_1} = \mathbf{Z_2}$, which may yield larger errors to $\mathbf{z}$, but have reduced influence on the accuracy of $\mathbf{q}$. This will be shown in the final results.

To make $\mathbf{z}$ more accurate, Eq. (7) is used to solve for $\mathbf{z_1}$ each step after $\mathbf{q_1}$ has been calculated, we call this the correction of $\mathbf{z}$. Substituting for $\mathbf{q_1}$ into Eq. (7) we calculate $\mathbf{z_1} = [1.1321,\ 2.2054]^T$. Let $t_0 = t_1$, $\mathbf{q_0} = \mathbf{q_1}$, $\mathbf{z_0} = \mathbf{z_1}$, we proceed to the next step where $t_1 = t_0 + h = 3.2995 - 0.0005 = 3.2990$. Upon the end of this second step, we have $\mathbf{q_1} = \mathbf{Q_2} = [0.8032,\ 0.7814,\ -0.5946]^T$ and $\mathbf{Z_2} = [1.1284,\ 2.1987]^T$. The procedure is repeated through each step until the iteration comes to step 314, when $\mathbf{q_1} = \mathbf{Q_2} = [0.6968,\ 1.0478,\ -0.6088]^T$ where q_2 has violated the joint limit of $p/3$, therefore, the integration returns and is initiated with a different initial configuration. The second iteration assumes $\mathbf{q_0} = [0.8516,\ 0.2157,\ 0.8078]^T$, and integrates till the intersection point C with $\mathbf{q}^c = [0.7556,\ 0.4152,\ 1.0037]^T$. Since $\mathbf{q}^c$ is far from the singular configuration $\mathbf{q}^s$, the initial configuration that admits a kinematically smooth configuration is calculated to be $[0.8516,\ 0.2157,\ 0.8078]^T$.

Having solved the problem by the original method, then optimization-based method is used to find the best initial configuration that will admit a kinematically smooth motion. For this example, we define a simple cost function that evaluates displacement of each joint away from its neutral position such that $cost = \left\| \mathbf{q} - \mathbf{q}^{\mathbf{N}} \right\|^2$ (16)

where $\mathbf{q}^{\mathbf{N}}$ represents the vector of neutral positions. For this example, $\mathbf{q}^{\mathbf{N}}$ is chosen to be $[0, 0, 0]^T$. To simplify the calculations, only the initial configuration is considered here towards calculating the cost function. This is a natural choice since the manipulator is moving continuously from the initial to the end configurations, which means the initial value of the cost function roughly reflects the values along the entire path.

RESULTS AND DISCUSSION

Tables 1 and 2 present the configuration $\mathbf{q}$, $\mathbf{Z}_2$, $\mathbf{z}$ as the manipulator is moving through one unsuccessful trial and the final successful trial of the example of Fig. 3 without optimization. Figs. 4 and 5 illustrate the movements of the manipulator accordingly.

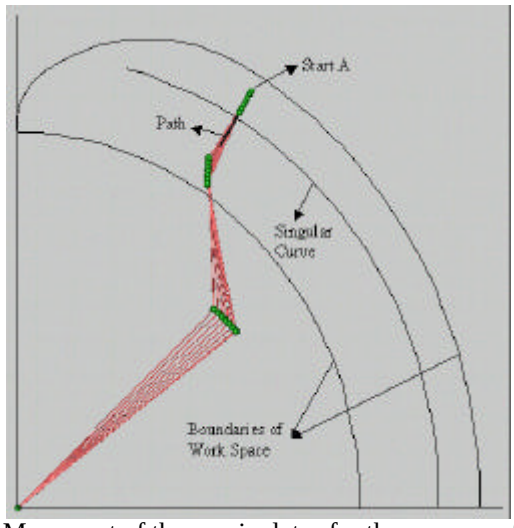

Fig. 4. Movement of the manipulator for the unsuccessful trial

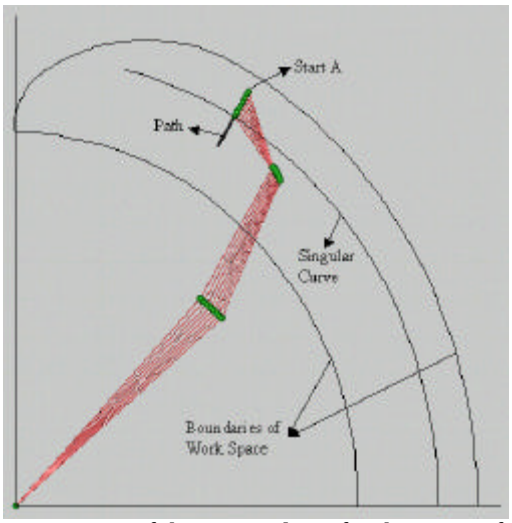

Fig. 5. Movement of the manipulator for the successful trial

For both cases, it can be observed that $\mathbf{q}$ is changing continuously, which means that the manipulator is admitting kinematically smooth motion. Moreover, observing the difference between the two positions**,** the movement is following the given path exactly. But during the unsuccessful trial, when the process comes to step 314, one joint variable q_2 has violated joint limit constraint at 1.0472. This halts the integration process and causes the loop to iterate back in search of a feasible initial configuration. In the successful trial, joint variables change slowly and keep within the joint limits. The process continues until an approporiate initial point is calculated. By correcting each step, $\mathbf{Z}_2$ converges to $\mathbf{z}$ very quickly, indeed, both Tables 1 and 2 show that $\mathbf{Z}_2$ begins to converge to $\mathbf{z}$ from the second step and keeps the convergence process thereafter. In fact, the numerical results suggest that the correction of $\mathbf{z}$ has no influence on the accuracy of the final result of $\mathbf{q}$. Table 3 lists the intermediate results for the successful trial while no correction of $\mathbf{z}$ at each step has been made, $\mathbf{z}$ is simply made to equal to $\mathbf{Z}_2$. Although during the integration process, $\mathbf{z}$ from Tables 2 and 3 have large differences, $\mathbf{q}$ from both methods are exactly same. Therefore, it is possible to solve the problem without correcting for $\mathbf{z}$, if computational time is an issue. Table 4 shows the summarized final results. The calculated start configuration is $[0.8516,\ 0.2157,\ 0.8078]^T$. When the manipulator comes to the intersection point, it has the configuration $[0.7556,\ 0.4152,\ 1.0037]^T$, which is different from the singular configuration $[0.7662,\ 1.0472,\ -1.0472]^T$, so the start configuration is accepted and the procedure terminates.

Table 5 presents results obtained from the optimization algorithm, where it is shown that properties of the original method are maintained, i.e., smooth movement, convergence of $\mathbf{Z}_2$ to $\mathbf{z}$, and strictly following the path. Table 6 lists the results obtained from the first method without optimization and the modified method with optimization. It can be seen through the comparison that better initial configurations with less bias to the point on the path have been calculated with the optimization algorithm. Moreover, the cost function on both the initial and intersection positions of the result with optimization are much less than those without optimization. It is concluded that using the cost function for the motion along the entire path starting from the calculated configuration by optimization would be less than that starting from the configuration calculated using the first method. Optimization indeed helps determine a best solution admitting a kinematically smooth motion and minimizing the specified cost function. From Table 6 it is also observed that the singular configuration has the largest value of the cost function as expected, since this singular configuration here actually has two joint variables at their limits, which significantly contributes to the value of the cost function. Fig. 6 shows the singular configuration at the intersection point and the neutral configuration used for calculating the cost function. Fig. 7 shows the movement of the manipulator from the results obtained with optimization. Figs. 8 and 9 illustrates configurations calculated from the original method without

optimization and those from the modified method with optimization.

During the optimization process, only the initial configuration is considered towards calculating the cost function and has a positive implication on the value of the cost function along the entire path trajectory. However, this is only true for very short paths. Generally, the configurations of at least several critical points on the path need to be used to measure the overall cost function.

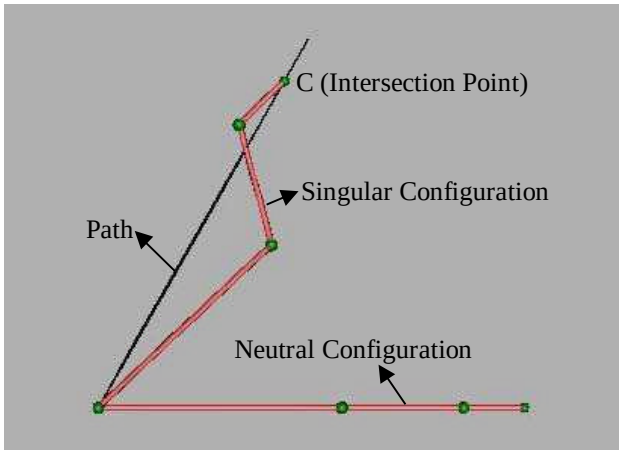

Fig. 6. Singular configuration and neutral configuration

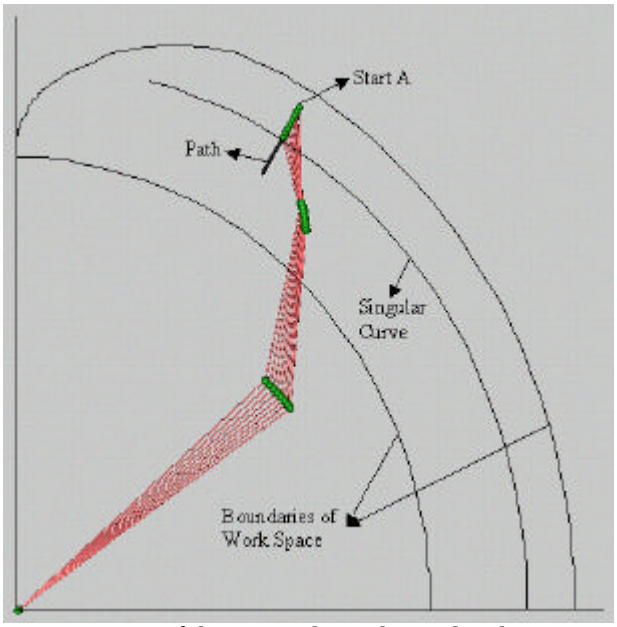

Fig.7. Movement of the manipulator obtained with optimization

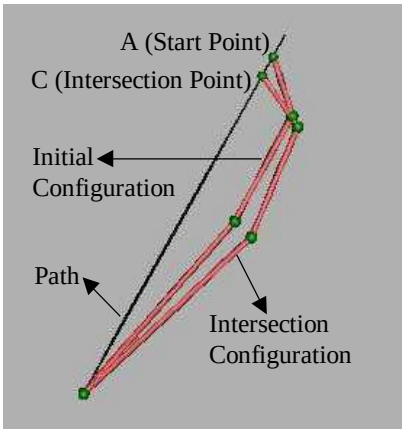

Fig. 8. Configuration obtained without optimization

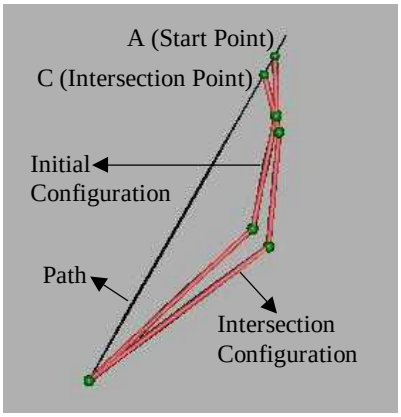

Fig. 9. Configuration obtained with optimization

CONCLUSION

A general method and accompanying formulation for designing kinematically smooth path trajectories of serial robot manipulators have been presented. The rigorous formulation is then implemented into code towards calculating an initial configuration (an inverse kinematic solution) of the arm that would admit a smooth motion throughout the path, without an interruption that is caused due to a switching of inverse solutions.

It was shown that the problem can be formulated as a set of constraints and governing equations as a system of differential-algebraic equations. It was also shown that well-established numerical methods called the Runge-Kutta of index 2 iterative numerical algorithm could be used to solve the problem. Optimization using a cost function that drives the arm towards an inverse solution was implemented. It was shown that an initial configuration for the starting point of the path could be readily calculated. Indeed, observations regarding the correction terms in the DAE index 2 solution were also made and the numerical algorithm was demonstrated in detail for a 3DOF example.

Step	t	q (Q2)			Z2		z		Real Position	Point on Path
0	3.3000	0.8030	0.7816	-0.5946			1.1276	2.1970	(3.2990,5.7137)	(3.3000,5.7158)
1	3.2995	0.8036	0.7803	-0.5945	-3.9613	-7.7424	1.1321	2.2054	(3.2995,5.7149)	(3.2995,5.7149)
2	3.2990	0.8032	0.7814	-0.5946	1.1284	2.1987	1.1284	2.1987	(3.2990,5.7140)	(3.2990,5.7140)
3	3.2985	0.8028	0.7824	-0.5947	1.1247	2.1920	1.1247	2.1920	(3.2985,5.7132)	(3.2985,5.7132)
50	3.2750	0.7844	0.8290	-0.6000	0.9727	1.9163	0.9727	1.9163	(3.2750,5.6725)	(3.2750,5.6725)
100	3.2500	0.7660	0.8754	-0.6039	0.8475	1.6883	0.8475	1.6883	(3.2500,5.6292)	(3.2500,5.6292)
150	3.2250	0.7485	0.9192	-0.6065	0.7487	1.5076	0.7487	1.5076	(3.2250,5.5859)	(3.2250,5.5859)
200	3.2000	0.7320	0.9606	-0.6081	0.6691	1.3613	0.6691	1.3613	(3.2000,5.5426)	(3.2000,5.5426)
250	3.1750	0.7161	1.0000	-0.6088	0.6036	1.2406	0.6036	1.2406	(3.1750,5.4993)	(3.1750,5.4993)
300	3.1500	0.7009	1.0376	-0.6089	0.5489	1.1395	0.5489	1.1395	(3.1500,5.4560)	(3.1500,5.4560)
313	3.1435	0.6971	1.0471	-0.6088	0.5361	1.1159	0.5361	1.1159	(3.1435,5.4447)	(3.1435,5.4447)
314	3.1430	0.6968	**1.0478**	-0.6088	0.5352	1.1141	0.5352	1.1141	(3.1430,5.4438)	(3.1430,5.4438)

Table 1. Traced results for one unsuccessful trial

Step	t	q (Q2)			Z2		z		Real Position	Point on Path
0	3.3000	0.8516	0.2157	0.8078			0.8982	1.7536	(3.3007,5.7151)	(3.3000,5.7158)
1	3.2995	0.8515	0.2162	0.8083	0.4802	0.8513	0.8961	1.7502	(3.2995,5.7149)	(3.2995,5.7149)
2	3.2990	0.8511	0.2168	0.8090	0.8937	1.7457	0.8937	1.7457	(3.2990,5.7140)	(3.2990,5.7140)
3	3.2985	0.8508	0.2174	0.8097	0.8912	1.7412	0.8912	1.7412	(3.2985,5.7132)	(3.2985,5.7132)
50	3.2750	0.8360	0.2465	0.8408	0.7898	1.5555	0.7898	1.5555	(3.2750,5.6725)	(3.2750,5.6725)
100	3.2500	0.8210	0.2763	0.8717	0.7048	1.3991	0.7048	1.3991	(3.2500,5.6292)	(3.2500,5.6292)
150	3.2250	0.8069	0.3051	0.9006	0.6366	1.2731	0.6366	1.2731	(3.2250,5.5859)	(3.2250,5.5859)
200	3.2000	0.7935	0.3330	0.9278	0.5808	1.1696	0.5808	1.1696	(3.2000,5.5426)	(3.2000,5.5426)
250	3.1750	0.7807	0.3602	0.9535	0.5343	1.0830	0.5343	1.0830	(3.1750,5.4993)	(3.1750,5.4993)
300	3.1500	0.7685	0.3867	0.9780	0.4951	1.0096	0.4951	1.0096	(3.1500,5.4560)	(3.1500,5.4560)
350	3.1250	0.7568	0.4126	1.0014	0.4615	0.9466	0.4615	0.9466	(3.1250,5.4127)	(3.1250,5.4127)
355	3.1225	0.7556	0.4152	1.0037	0.4585	0.9408	0.4585	0.9408	(3.1225,5.4083)	(3.1225,5.4083)

Table 2. Traced results for the successful trial

Step	t	q (Q2)			Z2 (z)		Real Position	Point on Path
0	3.3000	0.8516	0.2157	0.8078	0.8982	1.7536	(3.3007,5.7151)	(3.3000,5.7158)
1	3.2995	0.8515	0.2162	0.8083	0.4802	0.8513	(3.2995,5.7149)	(3.2995,5.7149)
2	3.2990	0.8511	0.2168	0.8090	1.3089	2.6434	(3.2990,5.7140)	(3.2990,5.7140)
3	3.2985	0.8508	0.2174	0.8097	0.4767	0.8446	(3.2985,5.7132)	(3.2985,5.7132)
50	3.2750	0.8360	0.2465	0.8408	1.1755	2.4042	(3.2750,5.6725)	(3.2750,5.6725)
100	3.2500	0.8210	0.2763	0.8717	1.0645	2.2052	(3.2500,5.6292)	(3.2500,5.6292)
150	3.2250	0.8069	0.3051	0.9006	0.9743	2.0432	(3.2250,5.5859)	(3.2250,5.5859)
200	3.2000	0.7935	0.3330	0.9278	0.8993	1.9086	(3.2000,5.5426)	(3.2000,5.5426)
250	3.1750	0.7807	0.3602	0.9535	0.8360	1.7951	(3.1750,5.4993)	(3.1750,5.4993)
300	3.1500	0.7685	0.3867	0.9780	0.7819	1.6979	(3.1500,5.4560)	(3.1500,5.4560)
350	3.1250	0.7568	0.4126	1.0014	0.7349	1.6139	(3.1250,5.4127)	(3.1250,5.4127)
355	3.1225	0.7556	0.4152	1.0037	0.1863	0.2755	(3.1225,5.4083)	(3.1225,5.4083)

Table 3. Traced results for the successful trial without **z** correction

	t	q			Real Position	Point on Path
Start	3.3000	0.8516	0.2157	0.8078	(3.3007,5.7151)	(3.3000,5.7158)
Intersection Point	3.1225	0.7556	0.4152	1.0037	(3.1225,5.4083)	(3.1225,5.4083)
Singular Configuration	3.1225	0.7662	1.0472	-1.0472	(3.1225,5.4083)	(3.1225,5.4083)

Table 4. Summarized final result without optimization

Step	t	q (Q2)			Z2		z		Real Position	Point on Path
0	3.3000	0.7615	0.5989	0.2234			0.6838	1.3966	(3.3005,5.7155)	(3.3000,5.7158)
1	3.2995	0.7611	0.5996	0.2238	0.6986	1.4172	0.6820	1.3933	(3.2995,5.7149)	(3.2995,5.7149)
2	3.2990	0.7608	0.6004	0.2241	0.6802	1.3867	0.6802	1.3900	(3.2990,5.7140)	(3.2990,5.7140)
3	3.2985	0.7604	0.6011	0.2245	0.6784	1.7412	0.6784	1.3867	(3.2985,5.7132)	(3.2985,5.7132)
50	3.2750	0.7445	0.6343	0.2405	0.6049	1.2490	0.6049	1.2490	(3.2750,5.6725)	(3.2750,5.6725)
100	3.2500	0.7285	0.6678	0.2567	0.5421	1.1308	0.5421	1.1308	(3.2500,5.6292)	(3.2500,5.6292)
150	3.2250	0.7133	0.6997	0.2722	0.4909	1.0342	0.4909	1.0342	(3.2250,5.5859)	(3.2250,5.5859)
200	3.2000	0.6988	0.7304	0.2870	0.4484	0.9536	0.4484	0.9536	(3.2000,5.5426)	(3.2000,5.5426)
250	3.1750	0.6849	0.7598	0.3013	0.4125	0.8855	0.4125	0.8855	(3.1750,5.4993)	(3.1750,5.4993)
300	3.1500	0.6716	0.7882	0.3151	0.3819	0.8271	0.3819	0.8271	(3.1500,5.4560)	(3.1500,5.4560)
350	3.1250	0.6588	0.8157	0.3284	0.3554	0.7764	0.3554	0.7764	(3.1250,5.4127)	(3.1250,5.4127)
355	3.1225	0.6576	0.8184	0.3297	0.3530	0.7717	0.3530	0.7717	(3.1225,5.4083)	(3.1225,5.4083)

Table 5.Traced optimized result with optimization

	t	$\mathbf{q}$			Real Position	Point on Path	Discomfort
Start (without optimization)	3.3000	0.8516	0.2157	0.8078	(3.3007,5.7151)	(3.3000,5.7158)	1.4243
Start (with optimization)	3.3000	0.7614	0.5982	0.2245	(3.3005,5.7155)	(3.3000,5.7158)	0.9883
Intersection Point (without optimization)	3.1225	0.7556	0.4152	1.0037	(3.1225,5.4083)	(3.1225,5.4083)	1.7507
Intersection Point (with optimization)	3.1225	0.6576	0.8184	0.3297	(3.1225,5.4083)	(3.1225,5.4083)	1.2109
Singular Configuration	3.1225	0.7662	1.0472	-1.0472	(3.1225,5.4083)	(3.1225,5.4083)	2.7803

Table 6. Comparison of results without and with optimization

ACKNOWLEDGMENTS

The research was funded by the US Army Tank Automotive Command (Automotive Research Center) and by the National Science Foundation under Grant No. 9983708.

REFERENCES

[1] Spanos, J. and Kohli, D., 1985, "Workspace analysis of regional structure of manipulators", *ASME J. of Mech. Trans. and Aut. in Design*, **107**, pp. 219-225.

[2] Shu, M., Kohli, D. and Dwivedi, S.H., 1986, "Workspaces and Jacobian surfaces of regional structures of industrial robots", *Proceedings of the 6th World Congress on Theory of Mach. and Mech.*, New Delhi, India, pp. 988-993.

[3] Lipkin, H. and Pohl, E., 1991, "Enumeration of singular configurations for robotic manipulators", *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **113,** pp. 272-279.

[4] Burdick, J.W., 1991, "A Classification of 3r regional manipulator singularities and geometries", *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, pp. 2670-2675.

[5] Pai, D.K. and Leu, M.C., 1992, "Generecity and singularities of robot manipulators", *IEEE Transactions on Robotics and Automation*, **8**, pp. 545-559.

[6] Lai, Z.C. and Yang, D.C.H., 1986, "A new method for the singularity analysis of simple six-link manipulators", *International Journal of Robotics Research*, **5**, No. 2, pp. 66-74.

[7] Tourassis, V.D. and Ang, M.H., 1992, "Identification and analysis of robot manipulator singularities", *International Journal of Robotics Research*, **11**, pp. 248-259.

[8] Soylu, R. and Duffy, J., 1988, "Hypersurfaces of special configurations of serial manipulators and related concepts. Part II: Passive joints, configurations, component manifolds and some applications", *Journal of Robotic Systems*, **5**, pp. 31-53.

[9] Nielsen, L., Dewit, C.C. and Hagander, P., 1991, "Controllability issues of robots in singular configurations", *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA.

[10] Oblak, D. and Kohli, D., 1988, "Boundary surfaces, limit surfaces, crossable and noncrossable surfaces in workspace of mechanical manipulators", *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **110**, pp. 389-396.

[11] Chevallereau, C. and Daya, B., 1994, "A new method for robot control in singular configurations with motion in any Cartesian direction", *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, Vol. 4, pp. 2692-2697.

[12] Chevallereau, C., 1996, "Feasible trajectories for a non redundant robot at a singularity", *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, pp. 1871-1876.

[13] Haug, E.J., Adkins, F.A., Qiu, C.C. and Yen, J., 1995, "Analysis of barriers to control of manipulators within accessible output sets", *Proceedings of the 20th ASME Design Engineering Technical Conference*, Boston, MA, Vol. 82, pp. 697-704.

[14] Haug, E.J., Luh, C.M., Adkins, F.A., and Wang, 1996, "Numerical algorithms for mapping boundaries of manipulator workspaces", *Transactions of the ASME Journal of Mechanical Design*, **118**, No. 1, pp. 228-234.

[15] Yeh, H.J., 1996, *Singularity and Workspace Analyses of Serial Robot Manipulators*. PhD Thesis, The University of Iowa, Iowa City, IA.

[16] Abdel-Malek, K. and Yeh, H.J., 1997, "Analytical boundary of the workspace for general three degree-of-freedom mechanisms", *International Journal of Robotics Research*, **16**, No. 2, pp. 198-213.

[17] Abdel-Malek, K., Adkins, F., Yeh, H.J., and Haug, E.J., 1997, "On the determination of boundaries to manipulator workspaces", *Robotics and Computer-Integrated Manufacturing*, **13**, No. 1, pp. 63-72.

[18] Abdel-Malek, K. and Yeh, H.J., 2000, "Crossable surfaces of robotic manipulators with joint limits", *ASME Journal of Mechanical Design*, **122**, pp. 52-60.

[19] Abdel-Malek, K., Yang, J. and Yeh, H.J., 2001, "Output control barriers of serial manipulators", *International Journal of Robotics and Automation*, **16**, No. 1, pp. 1-16.

[20] Nakamura, Y., 1991, *Advanced Robotics Redundancy and Optimization*, Addison-Wesley.

[21] Brasey, V. and Hairer, E., 1993, "Half-explicit Runge-Kutta methods for semi-explicit differential-algebraic equations of index 2, *SIAM J. Numer. Anal*., **30**, pp. 538-552.

[22] Hairer, E., Lubich, C. and Roche, M., 1989, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, Vol. 1409 of Lect. Notes in Math., Springer-Verlag, Berlin.

[23] Hairer, E. and Wanner, G., 1996, *Solving Ordinary Differential Equations II. Stiff and Differential-algebraic Problems*, Second Revised Edition, Springer, Berlin.

[24] Jay, L.O., 2001, *Preserving Poisson Structure and Orthogonality in Numerical Integration of Differential Equations*, Tech. Report, Dept. of Math., Univ. of Iowa, USA, Submitted.

[25] Murua, A., 1996, *Partitioned Runge-Kutta Methods for Semi-Explicit Differential-Algebraic Systems of Index 2*, Tech. Report EHU-KZAA-IKT-196, Univ. of the Basque Country.

[26] Jay, L.O., 2002, *Solution of Index 2 Implicit Differential-Algebraic Equations by Lobatto Runge-Kutta Methods*, Tech. Report, Dept. of Math., Univ. of Iowa, USA, Submitted.

[27] Jay, L.O., 1993, "Convergence of a class of Runge-Kutta methods for differential-algebraic systems of index 2, *BIT*, Vol. 33, pp. 137-150.