Proceedings of the 1999 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
September 19-23, 1999 • Atlanta, USA

Avoiding Discontinuities While Using the Minimum Infinity Norm to Resolve Kinematic Redundancy

Ian A. Gravagne and Ian D. Walker
Electrical and Computer Engineering, Clemson University
igravag@ces.clemson.edu, ianw@ces.clemson.edu

Abstract

Frequently in the practice of mechatronics, we see systems driven by multiple actuators where those actuators must work in a highly coupled fashion to achieve the desired results. In some cases, it may be desirable to provide more actuators than are strictly necessary, in which case the system becomes underdetermined, or redundant. Such underdetermined systems require the use of optimization schemes to resolve the redundancy in a manner consistent with a secondary task, such as the minimization of applied torques or expended energy. In a previous paper, we explored the ramifications of using a local optimization algorithm based on the least infinity norm. While a beneficial algorithm in many respects, it sometimes provides solutions which exhibit non-unique and discontinuous characteristics over time. In this paper, we propose one possible remedy for these problems, and continue to reveal more structure behind the least infinity norm and the infinity inverse, applying our results to redundancy resolution of a robot manipulator.

1 Introduction

The study of kinematic redundancy resolution is fundamentally a study in optimization techniques. Situations requiring the adjustment of a finite set of variables in accordance with a scalar constraint function abound, as do methods of sensibly combining the variables into a scalar constraint. However, simply choosing an optimization criteria does not necessarily provide insight into an obvious optimization procedure, and some side effects of using a given procedure may render it all but useless. In mechatronic systems, this finite set of variables often represent quantaties such as applied torques and forces, positions or velocities, input currents, etc. In the majority of these systems, these variables work in a coupled fashion, be it serially or in a closed chain. Here, an innocent question arises. Assume there are more variables than the minimum number strictly necessary to accomplish certain tasks (the system is redundant). If each variable exhibits upper limits (i.e. maximum speeds or maximum torques), can they "share" the work load for a given task in such a manner that the redundant system achieves the task, while the non-redundant system cannot without exceeding at least one actuator limit? The answer is often yes, though the solution procedure may be non-trivial, and for this question the tool which will provide our optimization constraint is the infinity norm. From the point of view of the joint variables, the infinity norm pays strict attention to the magnitude of the individual variables, rather than "lumping" them into an arbitrary optimization constraint.

Least infinity norm solutions are not yet widely used in mechatronics and robotics because of certain numerical difficulties. Previously, we demonstrated that such solutions may in fact exhibit discontinuities and points of non-uniqueness in time-varying systems [1]. This paper proceeds to suggest one possible modification which fixes this problem, introducing a predictive parameter which indicates how "close" an infinity norm solution is to a discontinuous point. Using this parameter, the algorithm is able to smoothly transition over the discontinuity, as shown by several computer simulations.

Without loss of generality in the results, we will choose a basic testbed for this paper, the rigid-link, redundant, serial robot. This choice also allows the work in this paper to dovetail nicely with key results in a previous paper [1]. A rigid-link robot is classified as redundant if the number of joints exceeds the minimum required by the dimension of the task space. For example, an industrial six-axis robot may be redundant if it only serves to position its end-effector in space, with disregard for the orientation, requiring only 3 degrees of freedom (DOF). Similarly, a 7 or greater DOF robot will be redundant performing any 6-DOF tasks. If the robot has n joints and the task requires m degrees of freedom, we may relate the robot end-effector location

0-7803-5038-3/99/$10.00

and orientation $\underline{x}$ to the joint variables $\underline{\theta}$ by[1]

$$\underline{x} = \underline{f}(\underline{\theta}) \tag{1}$$

with $\underline{x} \in \Re^m$, and $\underline{\theta} \in \Re^n$. Redundancy implies $m < n$, or $\underline{f} : \Re^n \rightarrow \Re^m$. Typically, $\underline{f}$ will be a non-linear function (often trigonometric) and impossible to invert in a redundant case. Therefore, rather than specifying desired positions, we may differentiate (1) to get

$$\dot{\underline{x}} = J(\underline{\theta})\dot{\underline{\theta}}, \quad J(\underline{\theta}) = \frac{d\underline{f}}{d\underline{\theta}} \tag{2}$$

and specify desired velocities. Note in (2) the matrix J is, in an underdetermined case, not square but rather $J \in \Re^{m \times n}$ with $m < n$ as mentioned above. Much literature exists on the subject of redundant robots [2][3], and some of their benefits include increased capability for avoiding obstacles and singularities.

2 The Least Infinity Norm and the Infinity Inverse

What is required to solve non-square, underdetermined system like (2) is a "generalized inverse". There are an infinite number of such inverses, with the most well-known being the pseudoinverse, such that the inverse of (2) would be

$$\dot{\underline{\theta}}^{(2)} = J^+ \dot{\underline{x}} \tag{3}$$

where $J^+ = J^T(JJ^T)^{-1}$ and $\dot{\underline{\theta}}^{(2)}$ is the optimal two-norm solution. (This type of solution is used extensively in robotics and comprises the heart of the conventional approach to redundancy resolution). More precisely, equation (3) solves

$$\min_{\dot{\underline{\theta}}} \left\| \dot{\underline{\theta}} \right\|_2 \text{ subject to } \dot{\underline{x}} = J\dot{\underline{\theta}} \tag{4}$$

Note that evaluating a pseudoinverse at some update rate along a given trajectory yields the joint velocity vector which has minimum two-norm at every point of evaluation. Expression (4) may be weighted with the robot's inertia matrix (for instance) to yield a trajectory which attempts to minimize its instantaneous kinetic energy [2][3]. However, while a pseudoinverse solution minimizes the total joint "energy", it does not guarantee that each individual joint speed is as small as possible. (The problem may be reformulated to minimize not just velocities, but torques, currents, or other parameters [8].) As such, the pseudoinverse can provide solutions where individual elements exceed the capabilities of the associated actuator.

In order to confront this problem, Deo employed the infinity norm as a minimization criterion [4]. The infinity norm of an arbitrary vector $\underline{\varepsilon}$ is defined as

$$\|\underline{\varepsilon}\|_\infty = \max\{|\varepsilon_1|, |\varepsilon_2|, ..., |\varepsilon_n|\}, \qquad \underline{\varepsilon} \in \Re^n \tag{5}$$

In our case, we would want to solve

$$\min_{\dot{\underline{\theta}}} \left\| \dot{\underline{\theta}} \right\|_\infty \text{ subject to } \dot{\underline{x}} = J\dot{\underline{\theta}} \tag{6}$$

to minimize the maximum joint speed. Denoting the optimal solution as $\dot{\underline{\theta}}^{(\infty)}$, another way to think of the solution vector is to note that, if any maximum-magnitude element of $\dot{\underline{\theta}}^{(\infty)}$ exceeds the associated joint limit, then *it is not possible to achieve the desired task given the current joint variable limits.* In other words, least infinity norm solutions attempt to distribute the work between all available resources, minimizing each individual joint contribution as far as possible.

Unfortunately, while minimum infinity norm solutions serve as valuable tools, they are non-trivial to compute. Cadzow presented the first algorithm to find a minimum infinity norm solution to an underdetermined problem [5], and several alternative algorithms have subsequently appeared, offering varying degrees of stability and computational load [6][7]. Although the complexity of these numerical algorithms tends to thwart analysis efforts which concentrate on the structure of a problem, we have found that the solution to (6) can be formulated as

$$\dot{\underline{\theta}}^{(\infty)} = J^\# \dot{\underline{x}} \tag{7}$$

where $J^\# = Q(JQ)^{-1}$. This is the "infinity inverse", and its structure is very close to that of a pseudoinverse. The matrix Q may take many different forms, but one form proving particularly useful fills Q with only elements of the set $\{1, -1\}$ [1]. We now mention a very important property of minimum infinity norm solutions [5], used extensively in the algorithms which compute such solutions.

Equal Magnitude Property: The minimum infinity norm solution vector to an underdetermined system must have $n - m + 1$ elements *equal to the maximum magnitude* $\left\| \dot{\underline{\theta}} \right\|_\infty$. Only $m-1$ elements may be uniquely assigned with lesser magnitudes.

Though the infinity inverse does not lend itself to direct computation [1], the structure has well-defined

[1] A note on notation: vector quatities and functions will be denoted with underscores (for example, $\underline{x}$), matrices will be capitalized (J), and all other quantities (m, n) are scalar unless denoted otherwise. Sets and subsets are bold ($\mathbf{A}, \mathbf{K}$).

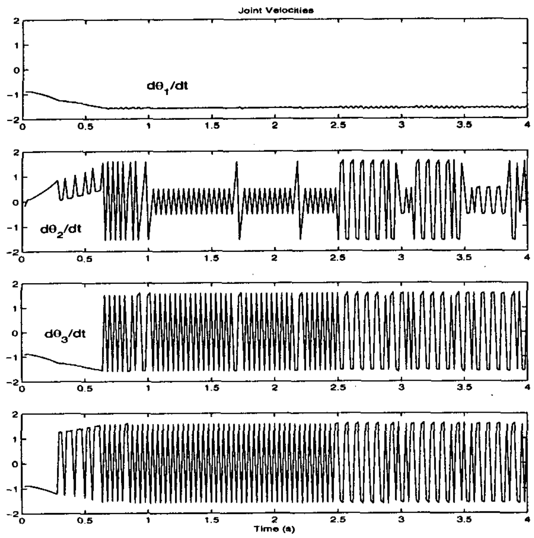

Figure 1: Joint velocities from the robot circle trajectory shown in figure (2). Note how the Equal Magnitude Property is always preserved.

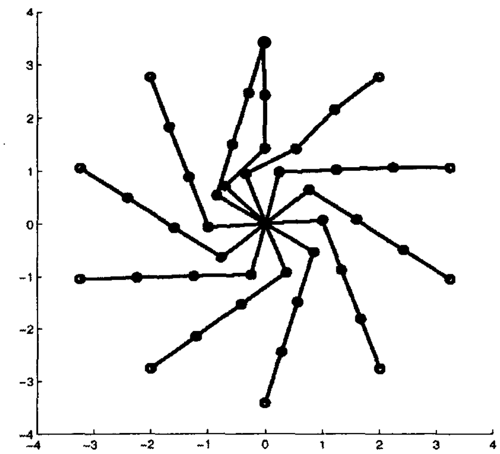

Figure 2: A picture of the four-link planar robot traversing a circle. The inital configuration is [135, -90, 45, 0] in CCW degrees from the horizontal. The robot end-effector travels CW.

geometric and algebraic meaning, and for any algorithmic solution $\underline{\dot{\theta}}^{(\infty)}$, there are a set of trivial rules to create the corresponding Q which computes that solution. With these tools in hand, it has been shown [1] that some situations exist where $\underline{\dot{\theta}}^{(\infty)}$ is not a unique solution (whereas the pseudoinverse always guarantees unique solutions). Furthermore, as the robotic system evolves and changes over time, if it moves "over" a point of non-uniqueness this can cause a discontinuity in the computed solutions. With a redundant robot, we can see an extreme case of this if we review an example of a 4-link planar robot. We take x-y positioning of the end-effector as the taskspace, so the robot has a 2-dimensional nullspace (2 degrees of redundancy). In figure 2, it attempts to execute a clockwise circle. Naturally, the algorithm attempts to reduce the maximum joint speed as far as possible, which soon leads to joint 1 (the center joint) doing all of the work. A pseudoinverse here would simply drop the speeds of joints 2,3 and 4 to zero, but the least infinity norm solutions must obey the "Equal Magnitude Property", and therefore at least two joints always oscillate between $-\left\|\underline{\dot{\theta}}\right\|_\infty$ and $+\left\|\underline{\dot{\theta}}\right\|_\infty$, as seen in figure 1. Again, this represents an extreme case of discontinuity using least infinity-norm solutions.

3 Tackling the Discontinuity Problem

To repair the discontinuity problem, we are interested in a predictive method which "senses" when a discontinuity is near, and gradually takes corrective action. Though not the only possible remedy, this method preserves those features of just-in-time local optimization which are most desirable, for instance, fast response to dynamically changing environments.

We now present our predictive method, which we have dubbed "rate mixing". It has the following form:

$$\underline{\dot{\theta}}^* = r\underline{\dot{\theta}}^{(\infty)} + (1-r)\underline{\dot{\theta}}^{(2)}, \quad 0 \le r \le 1 \tag{8}$$

A quick substitution of $\underline{\dot{\theta}}^*$ into (2) demonstrates that $\underline{\dot{\theta}}^*$ is a valid solution. Conceptually, we now want to vary this parameter r, called the "mixing factor", so that it stays near 1 most of the time, moving to 0 at points of discontinuity. Under these constraints, if r varies continuously, $\underline{\dot{\theta}}^*$ will vary continuously because $\underline{\dot{\theta}}^{(2)}$ is always unique and continuous. (The pseudoinverse solution need not be the only choice here, any sub-optimal continuous solution will work.) The choice of a good mixing factor depends on several conditions which we explore next.

To proceed with the discussion on how to deal with discontinuities, one should understand the geometry of least infinity norm problems. As mentioned earlier, we typically start in n-space ($\Re^n$) with a set of n joint variable constraints. Call this constraint set $\mathbf{C}_0$, defined in our case as

$$\mathbf{C}_0 = \left\{\dot{\theta}_i : -k_i \leq \dot{\theta}_i \leq k_i\right\} \quad i = 1, 2, ..., n \qquad (9)$$

and note that $\mathbf{C}_0 \subseteq \Re^n$, and takes the shape of a polytope aligned with the elementary basis vectors $[1, 0, 0, ...0]$, $[0, 1, 0, ...0]$, ..., $[0, 0, ..., 1]$. (If some actuator constraints are dependent, then the polytope may not align with certain basis directions.) If we make a variable substitution, $\tilde{\dot{\underline{\theta}}} = D\dot{\underline{\theta}}$, in the vast majority of cases, there exists a non-singular weighting matrix D such that we may normalize polytope $\mathbf{C}_0$ into a hypercube $\mathbf{C}_1$ with

$$\mathbf{C}_1 = \left\{\tilde{\dot{\theta}}_i : -k \leq \tilde{\dot{\theta}}_i \leq k\right\} \quad i = 1, 2, ..., n \qquad (10)$$

Letting $\tilde{J} = JD^{-1}$, the optimization problem can be solved in normalized space, and the solution then denormalized at the end, $\dot{\underline{\theta}}^{(\infty)} = D^{-1}\tilde{\dot{\underline{\theta}}}^{(\infty)}$. Henceforth in our discussion we assume the problem has already been normalized, and the actuator constraint set looks like $\mathbf{C}_1$, a hypercube in n dimensions.

From a geometric point of view, finding $\dot{\underline{\theta}}^{(\infty)}$ involves "growing" the hypercube (i.e. increasing k from 0) until one edge of it touches the solution space $\mathbf{S}$ [8], where $\mathbf{S} = \{\dot{\underline{\theta}} : J\dot{\underline{\theta}} = \dot{\underline{x}}\}$ and $\mathbf{S}$ is always parallel to the nullspace of J, denoted $\mathbf{N}$. Note $\mathbf{N} \in \Re^{n-m}$, and we will represent the nullspace with a matrix N such that the orthogonal columns of N span the nullspace. In other words, let

$$N \in \Re^{n \times (n-m)} : span\left\{\underline{n}_1, \underline{n}_2, ..., \underline{n}_{n-m}\right\} = \mathbf{N} \qquad (11)$$

As the hypercube expands, just where it first intersects the solution space is the point $\dot{\underline{\theta}}^{(\infty)}$. In the case of the two norm, we start not with a hypercube, but a hypersphere, and $\dot{\underline{\theta}}^{(2)}$ is the point where the sphere first touches $\mathbf{S}$. At this point, there can only be one solution because the hypersphere is "round", but in the case of a hypercube, if the nullspace is exactly aligned with a side or an edge of the hypercube, then $\mathbf{C}_1 \cap \mathbf{S}$ contains an infinity of solutions. This represents the case of a non-unique optimal solution $\dot{\underline{\theta}}^{(\infty)}$. Furthermore, if a time-varying system encounters this point in its trajectory, the solution could "hop" across the side of the hypercube, resulting in a discontinuity. It is important to note that this condition, which we will call the "Zero Subspace Angle" condition, is *necessary* but *not sufficient* for a discontinuity. We now proceed to define this condition more rigorously.

The *subspace angle* is defined geometically as the angle between two hyperplanes (subspaces $\mathbf{S}_1$ and $\mathbf{S}_2$) embedded in a higher dimensional space. For instance, if $span\{[0, 1, 0]^T\} = \mathbf{S}_1$ and $span\{[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T, [1, 0, 0]^T\} = \mathbf{S}_2$, then $angle(\mathbf{S}_1, \mathbf{S}_2) = 45°$. More precisely, let S_1 be a matrix with orthogonal columns which span $\mathbf{S}_1$, and S_2 be a matrix with orthogonal columns which span $\mathbf{S}_2$. Then,

$$\begin{gathered} angle(\mathbf{S}_1, \mathbf{S}_2) = \cos^{-1}(\sigma), \\ \sigma = \min\{diag(\Sigma)\} \quad \text{with } U\Sigma V^T = S_1^T S_2 \end{gathered} \qquad (12)$$

The subspace angle is the inverse cosine of the minimum singular value of $S_1^T S_2$ [9].

This is relatively unwieldy, but its use will be simplified shortly. We begin by assigning $S_1 = N$. (Because $\mathbf{S}$ and $\mathbf{N}$ are parallel linear spaces, the subspace angle will be the same whether the columns of S_1 span the solution space or the nullspace). Since we are concerned with the solution space intersecting edges and faces of the hypercube (which are always aligned with the elementary basis vectors), we must choose m of these basis vectors to be the columns of S_2, so that, if $\mathbf{N}$ and $\mathbf{S}_2$ are independent spaces, $\mathbf{N} \cup \mathbf{S}_2 \subseteq \Re^n$. That is, if the spaces are independent, then their union should be the complete space $\Re^n$. Now the question is, which m basis vectors should form $\mathbf{S}_2$? There will be $\binom{n}{m}$ possible arrangements and, from the above discussion, we want to know if the nullspace has a zero subspace angle with any one of those choices. Now we note the following result:

Zero Subspace Angle Condition For an underdetermined system employing least infinity norm optimization with an optimum solution $\dot{\underline{\theta}}^{(\infty)}$: $\dot{\underline{\theta}}^{(\infty)}$ non-unique $\Rightarrow \exists$ some $\mathbf{S}_2$ spanned by elementary basis vectors such that $\cos^{-1}(\sigma_1) = 0$.

Finding the subspace angle for $\binom{n}{m}$ subspaces could take a long time using the singular value decomposition method, and yields little intuition. Since we do not need the actual angle, only a number representative of the angle (in a one-to-one correspondence) we may note that, by filling a $n \times n$ matrix with columns which span the two subspaces, a zero determinant indicates that the column space is not complete. Since we know that, by construction, S_2 consists of m columns of 0's and 1's (no column repeating), it will always span $\Re^m$. Therefore, to find the minimum subspace angle over all possible faces and edges of the hypercube (all possible S_2), we may concatenate N and S_2 into a $n \times n$ matrix and find its determinant. A zero determinant corresponds to a zero minimum subspace angle; larger absolute value determinants indicate larger subspace angles. Note that we may also swap rows in this concatenation matrix in the following manner:

$$[N \mid S_2] \rightarrow \left[\begin{array}{cc} N_1 & 0 \\ N_2 & I_{m\times m} \end{array} \right] \tag{13}$$

with the result that

$$|\det [N \mid S_2]| = |\det[N_1] * \det[I]| = |\det[N_1]| \tag{14}$$

Now $\binom{n}{m}$ subspace angle calculations have been distilled down to $\binom{n}{m}$ determinants of size $(n-m)\times(n-m)$. Define

$$d_{\min} = \min_{\mathbf{S}_2} \{|\det[N_1]|\} \tag{15}$$

In an evolving trajectory, if $d_{\min}$ approaches zero, the solution $\underline{\dot{\theta}}^{(\infty)}$ gets closer and closer to a point of non-uniqueness, and possible discontinuity. Note $d_{\min}$ is continuous (although not smooth) if the elements of N vary continuously; this implies the elements of J varying continuously – a reasonable expectation if the input trajectory contains no discontinuities.

Only one step remains to refine $d_{\min}$ into the mixing factor r from (8). We must limit $d_{\min}$ to exist only between 0 and 1, which can be accomplished as

$$r = 1 - e^{-a \cdot d_{\min}} \tag{16}$$

Recall that $d_{\min} > 0$. The parameter a adjusts how quickly the algorithm switches from one solution to the other, tending to "round out" the sharp edges of the hypercube.

4 Results Using Rate Mixing

The rate mixing method has proven quite effective in a number of examples. Figure 3 shows a revised version of the circle example. The first plot shows how r varies with time, heavily favoring the pseudoinverse solution to avoid the oscillations present from figure 1. The picture of the actual robot looks to the naked eye identical to figure 2, and is not reproduced here. Figure 4 shows a linear trajectory, specifying $\underline{\dot{x}} = [-0.8, -0.8]$ for about 4 seconds. The robot picture does yield much insight, and is not included; note that this trajectory contains an isolated discontinuity which the robot smoothly avoids, favoring the infinity norm solution most of the time. Also note that the mixing factor hits zero once where there is no discontinuity. This represents a point of non-uniqueness, however the robot's trajectory did not take it "across" the non-unique point to create a discontinuity. This instance reflects the necessary, but not sufficient, quality of the mixing factor.

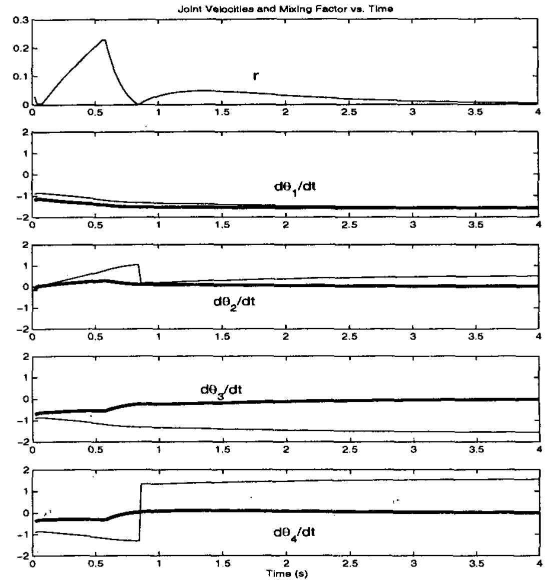

Figure 3: The "corrected" circle example. In the four velocity plots, the thick line represents the mixed solution, the thin line is what the infinity inverse provides on its own; mixing factor r is shown at the top.

5 Remarks on Computation and Special Cases

The computational load involved in producing $\underline{\dot{\theta}}^*$ at any reasonable sampling rate could grow quite large. However, careful construction of the algorithm can help reduce this load. One notably stable algorithm for finding $\underline{\dot{\theta}}^{(\infty)}$ involves computing $\underline{\dot{\theta}}^{(2)}$ as the first step [7]. Additionally, a modified computation of the pseudoinverse J^+ in this algorithm using singular value decomposition yields the nullspace N with little additional effort. For rigid-link robots, the typical degree of redundancy will be only one or two; in the most common case – one degree redundant – the determinants in (15) boil down to a simple search for the smallest magnitude element of the nullspace vector. In addition, modern computers executing these types of algorithms typically provide time to spare even on very complex computations. The initial step in the rate-mixing algorithm is the computation of the pseudoinverse. In the case of 7-DOF robot with a 6-DOF taskspace, an iterative singular value decomposition (calculating U, Σ and V) takes an average of only 190μs on a Pentium II 450Mhz CPU, with the maximum time-to-converge around 210μs. (This experiment was performed in a real-time operating system and averaged over 10 000

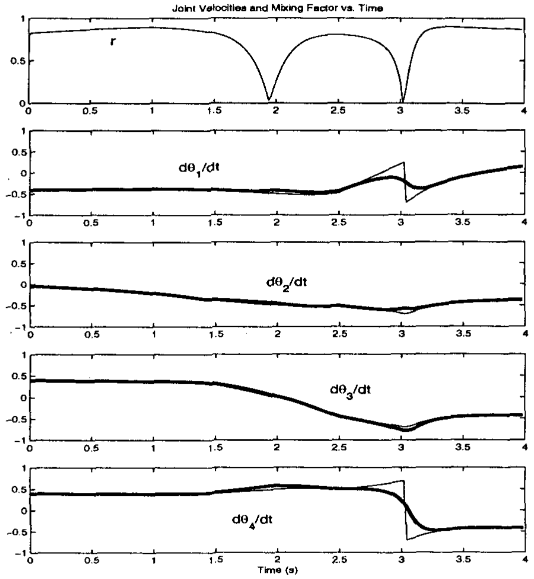

Figure 4: A linear trajectory. The four link planar robot of figure (2) starts with initial position [135,-45,45,0] in CCW degrees from the horizontal. The thick line is the final result, the thin line is the least ∞-norm solution alone.

randomly generated 6×7 matrices.)

Of mathematical interest is the following observation. In a one-degree redundant case, the nullspace is one dimensional and it is easy to show that any arbitrary minimum p-norm solution to (2) ($p > 1$) must be a linear combination of $\underline{\dot{\theta}}^{(\infty)}$ and $\underline{\dot{\theta}}^{(2)}$. Since the rate mixing expression does in fact provide a convex combination of the two solutions, it is likely that the resulting $\underline{\dot{\theta}}^{*}$ closely follows a minimum p-norm solution, with $2 \ll p$.

6 Conclusions

In this work, we have reviewed the benefits of minimizing the infinity norm of a solution to an underdetermined system. We noted that the presence of discontinuities presents one of the current challenges while using infinity norm solutions in time-varying systems. In order to smooth out the solutions, while still preserving the essence of a least infinity norm answer, we suggested the rate mixing solution, based on the geometry of the problem. This allows for a varying mixing-factor r to switch from infinity norm solutions to sub-optimal continuous solutions near discontinuities, with the mixing factor being a relative indication of the minimum subspace angle. The simulator results demonstrate success at avoiding discontinuities in typical trajectories. Though we chose a simple test case to frame our results, they represent a general method for solving dicontinuity problems with infinity norm optimization and could be used in any number of different situations, including robot redundancy resolution at the joint velocity and torque levels[2].

References

[1] I. Gravagne and I.D. Walker, "Properties of Minimum Infinity-Norm Optimization Applied to Kinematically Redundant Robots," Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS) October 1998, Victoria, Canada

[2] B. Siciliano, "Kinematic Control of Redundant Robot Manipulators: a Tutorial," Journal of Intelligent and Robotic Systems, No. 3, pp.201-212, 1990

[3] D.N. Nenchev, "Redundancy Resolution Through Local Optimization: a Review," Journal of Robotic Systems, Vol. 6, No. 6, pp.769-798, 1989

[4] A.S. Deo, I.D. Walker, "Minimum Effort Inverse Kinematics for Redundant manipulators," IEEE Trans. Robotics and Automation, Vol. 13, No. 5, pp. 767-775

[5] J.A. Cadzow, "An Finite Algorithm for the minimum ℓ_∞-norm Solution to a System of Consistent Linear Equations," SIAM Journal of Numerical Analysis, Vol. 10, No. 4, pp.607-617, 1973

[6] J.A. Cadzow, "An Efficient Algorithmic Procedure for obtaining a minimum ℓ_∞-norm Solution to a System of Consistent Linear Equations," SIAM Journal of Numerical Analysis, Vol. 11, No. 6, pp.1151-1165, 1974

[7] L. Kolev, "An Algorithm of Finite Number of Iterations for the Minimum Fuel and Minimum Amplitude Control Problems," Int'l Journal of Control, Vol 22, No. 1, pp.97-102, 1975

[8] I.C. Shim and Y.S. Yoon, "Stability Constraint for Torque Optimization of a Redundant Manipulator," Proc. IEEE Conf. Robotics and Automation (ICRA) April 1997, Albuquerque, NM

[9] Chatelin, Eigenvalues of Matrices, pub. John Wiley & Sons, 1993, pp.44-45

[2] This work has been supported by NSF grant CMS-9796328; and DOE contract DE-FG07-97ER14830