A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators

Tan Fung Chan and Rajiv V. Dubey

***Abstract*—It is proposed to use weighted least-norm solution to avoid joint limits for redundant joint manipulators. A comparison is made with the gradient projection method for avoiding joint limits. While the gradient projection method provides the optimal direction for the joint velocity vector within the null space, its magnitude is not unique and is adjusted by a scalar coefficient chosen by trial and error. It is shown in this paper that one fixed value of the scalar coefficient is not suitable even in a small workspace. The proposed manipulation scheme automatically chooses an appropriate magnitude of the self-motion throughout the workspace. This scheme, unlike the gradient projection method, guarantees joint limit avoidance, and also minimizes unnecessary self-motion. It was implemented and tested for real-time control of a seven-degree-of-freedom (7-DOF) Robotics Research Corporation (RRC) manipulator.**

I. Introduction

Redundant joints allow a manipulator to avoid joint limits, singularities, and obstacles, while following a desired end-effector trajectory. Redundancy is also utilized to minimize joint velocities or actuator torques. In this paper we focus on avoiding joint limits, which is necessary for uninterrupted operation of the manipulator.

Redundancy utilization for a particular task such as joint limit avoidance is generally achieved through global or local optimization of a performance criterion while satisfying the kinematic equations relating the end-effector trajectory to the joint trajectory [1]–[10]. Since global optimization requires complete trajectory information in advance, it is not well suited for tasks requiring continuous trajectory modification based on sensory feedback. Local optimization schemes determine a joint trajectory such that at each point along the end-effector trajectory, a performance criterion is locally optimized. Even though local optimization schemes may not result in the "best" joint trajectory for a complete operation, they are most suitable for on-line implementation.

Gradient projection method (GPM) [2] has been widely used in the literature for utilizing redundancy to avoid joint limits. A performance criterion is defined as a function of joint limits and its gradient is projected onto the null space projection matrix of the Jacobian to obtain the self-motion necessary to optimize the performance criterion. However, there are several problems associated with the gradient projection method. These include the selection of the appropriate scalar coefficient that determines the magnitude of the self-motion, unnecessary self-motion, oscillations [11] in the joint trajectory, and in many cases, inability to avoid the joint limits in time to avoid disruption.

Weighted least-norm (WLN) solution was originally suggested by Whitney [1] to resolve redundancy. It has been utilized for minimizing energy by using the inertia matrix as the weighting matrix. Hollerbach and Suh [4] used it for minimizing joint torques.

Manuscript received October 12, 1992; revised August 19, 1993. This work was supported by the NASA Langley Research Center under Grant NAG-1-1008 and Sandia National Laboratory under Grant AD-0623. This paper was partially presented at the Intentional Conference of Robotics and Automation, Atlanta, GA, May 2–6, 1993.

The authors are with the Mechanical and Aerospace Engineering Department, The University of Tennessee, Knoxville, TN 37996-2210 USA.

IEEE Log Number 9403334.

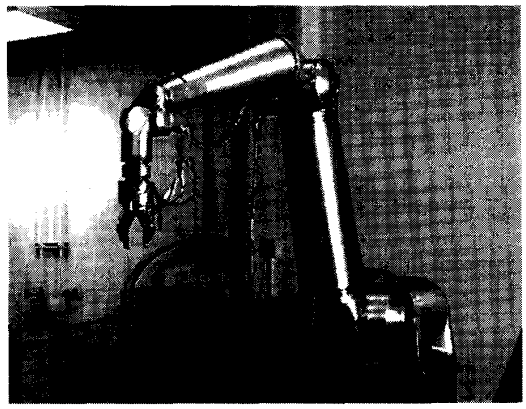


Fig. 1. RRC K-2107 manipulator.

In this paper we present a scheme for avoiding joint limits using the WLN solution. A comparison is made with the GPM. Using several examples, advantages of the proposed WLN method over the GPM are presented for avoiding joint limits. Both schemes are implemented in real-time on a 7-DOF Robotics Research Corporation (RRC) manipulator (Fig. 1).

II. Local Optimization Using Weighted Least-Norm Solution and Gradient Projection Method

The kinematic equation describing the relationship between the end-effector velocity and joint velocities is given as:

$$\underline{\dot{x}} = J\underline{\dot{\theta}} \tag{1}$$

where $\underline{\dot{x}}$ is an $m \times 1$ vector consisting of three linear and three rotational velocity components; $\underline{\dot{\theta}}$ is an $n \times 1$ vector consisting of the joint velocities; and J is an $m \times n$ Jacobian matrix.

For a redundant manipulator, we have $m < n$; therefore, an infinite number of joint velocity vectors $\underline{\dot{\theta}}$ exist for a given $\underline{\dot{x}}$ in (1). The joint velocity vectors can be computed as:

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

where J^{+} is the Moore-Penrose inverse [12] of the Jacobian; $J^{+}\underline{\dot{x}}$ is the least-norm (LN) solution of (1), i.e., it provides the $\underline{\dot{\theta}}$ with minimum Euclidian norm; $(I - J^{+}J)\underline{\phi} \in \mathcal{N}(J)$, the null space of J, is the homogeneous solution of (1), and $\underline{\phi} \in R^{n}$ is an arbitrary vector. The homogeneous solution is referred to as the self-motion of the manipulator and does not cause any end-effector motion.

A. Weighted Least-Norm (WLN) Solution

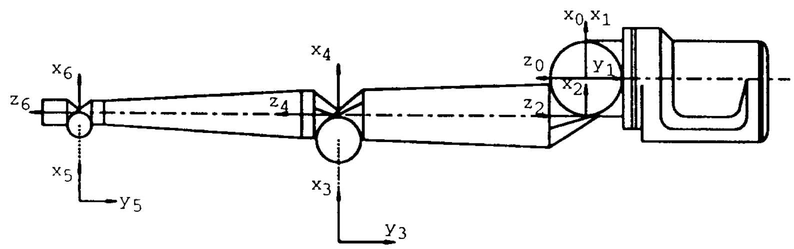
In order to penalize the motion of some joints over the others, we define the weighted norm of the joint velocity vector $\underline{\dot{\theta}}_{W}$ as follows:

$$|\underline{\dot{\theta}}|_{W} = \sqrt{\underline{\dot{\theta}}^{T} W \underline{\dot{\theta}}} \tag{3}$$

where $W \in R^{n \times n}$ is a symmetric and positive definite weighting matrix. In most cases this will be taken as a diagonal matrix for simplicity.

1042–296X/95$04.00 © 1995 IEEE

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.



α_i (degree)	a_i (inch)	d_i (inch)	$\theta_{imin} < \theta_i < \theta_{imax}$ (degree)
-90	0	0	$-180 < \theta_1 < 180$
90	-5.625	0	$-45 < \theta_2 < 135$
-90	-4.25	37.985	$-180 < \theta_3 < 180$
90	4.25	0	$-180 < \theta_4 < 0$
-90	-1.937	37.996	$-360 < \theta_5 < 360$
90	1.937	0	$-180 < \theta_6 < 0$
0	0	10.619	$-720 < \theta_7 < 720$

Fig. 2. DH table of RRC K-2107 manipulator.

Now for the purpose of analysis we introduce the following transformations:

$$J_W = JW^{-1/2} \quad \text{and} \quad \underline{\dot{\theta}}_W = W^{1/2}\underline{\dot{\theta}}. \tag{4}$$

Using the above transformations, we can rewrite (1) as

$$\underline{\dot{x}} = J_W \underline{\dot{\theta}}_W \tag{5}$$

and (3) as

$$|\underline{\dot{\theta}}|_W = \sqrt{\underline{\dot{\theta}}_W^T \underline{\dot{\theta}}_W} \tag{6}$$

The LN solution of (5) is the following:

$$\underline{\dot{\theta}}_W^* = J_W^+ \underline{\dot{x}}. \tag{7}$$

Thus, using the second part of (4), the weighted least-norm solution of (1) can be obtained by:

$$\underline{\dot{\theta}}_{W_m} = W^{-1/2}\underline{\dot{\theta}}_W^*. \tag{8}$$

From the definition of pseudoinverse, the WLN solution can be shown to be the following:

$$\underline{\dot{\theta}}_{W_m} = W^{-1}J^T[JW^{-1}J^T]^{-1}\underline{\dot{x}} \tag{9}$$

provided J is full ranked. Since the joint velocity vector $\underline{\dot{\theta}}_W^*$ is the LN solution of (5), the method proposed in [14] can be used to reduce calculations. In this method, particular and homogeneous solutions of (5) are determined; and then, the particular solution's component along the homogeneous solution is subtracted from the particular solution to obtain the LN solution.

B. Gradient Projection Method (GPM)

In order to improve a performance criterion $H(\underline{\theta})$ using the GPM [2], the redundancy is resolved by substituting $k\nabla H(\underline{\theta})$ for $\underline{\phi}$ in (2):

$$\underline{\dot{\theta}} = J^+\underline{\dot{x}} + k(I - J^+J)\nabla H \tag{10}$$

where k is a real scalar coefficient. It is taken to be positive if $H(\underline{\theta})$ is to be maximized and negative if $H(\underline{\theta})$ is to be minimized. The maximum allowable value of k should be limited by bounds on joint velocities [13] or actuator torques.

III. Performance Criteria for Avoiding Joint Limits

In order to avoid joint limits using redundancy, a performance criterion as a function of joint angles and their limits may be defined and optimized. Liegeois [2] in his paper on GPM, minimized the following performance criterion to avoid joint limits:

$$H(\underline{\theta}) = \frac{1}{6}\sum_{i=1}^{6}\left(\frac{\theta_i - a_i}{a_i - \theta_{i,\max}}\right)^2$$

$$a_i = \frac{1}{2}(\theta_{i,\max} + \theta_{i,\min}) \tag{11}$$

where θ_i is the ith joint angle, $\theta_{i,\max}$ and $\theta_{i,\min}$ are the upper and lower limits, respectively, on the joint angle θ_i. Since from the value of this performance criterion, it is not possible to determine if a joint has reached its limit, it is not well suited for on-line control. Zghal *et al.* [15] have proposed the performance criterion to avoid the joint limits to be in the following form:

$$H(\underline{\theta}) = \sum_{i=1}^{n}\frac{1}{4}\frac{(\theta_{i,\max} - \theta_{i,\min})^2}{(\theta_{i,\max} - \theta_i)(\theta_i - \theta_{i,\min})}. \tag{12}$$

Note that $H(\underline{\theta})$ automatically gives higher weight to the joints nearing their limits and goes to infinity at the joint bounds. It is also normalized to account for the variations in the motion range. Therefore $H_i(\underline{\theta}_i)$, the ith term inside the summation in (12), is equal to 1 in the middle of the joint limits, and goes to infinity at the limits. Fig. 3 shows the plot of the second term, $H_2(\underline{\theta}_2)$, as a function of θ_2.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.

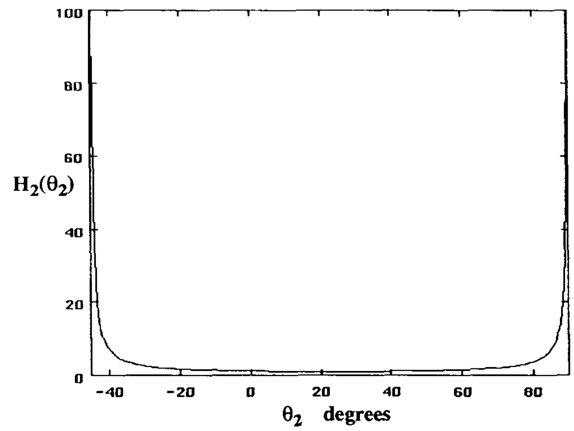


Fig. 3. Performance criterion $H_2(\theta_2)$.

IV. Problems with Using the Gradient Projection Method for Avoiding Joint Limits

The GPM has been widely used in the literature for avoiding joint limits. Performance criteria such as shown in (11) and (12) are optimized using the GPM in (10). The GPM provides only the direction of the homogeneous solution component to be used for optimization, and the magnitude of the homogeneous solution depends on the magnitude of the projection of the gradient onto the null space projection matrix and the scalar coefficient k in (10). This creates the following implementation problems for real-time control:

1) Since k is chosen based on trial and error, or based on joint velocity or torque limits, it may be too large or too small. If it is too large, it results in oscillations as was shown by Euler *et al.* [11]. If k is too small, it will be able to change the manipulator configurations only when the projection vector $(I - J^+J)\nabla H$ becomes large, which may be too late to avoid joint limits in time because of the hardware bounds on joint velocities and accelerations. As shown later by examples, most often a k value suitable for a given configuration is either too large or too small for other configurations.
2) If gain k is sufficiently large for quick optimization, GPM results in configurations that are near optimum, which corresponds to joint angles in the middle of their ranges. Keeping the joints right in the middle is not necessary as long as they are far enough from the bounds. It, in fact, results in excessive and unnecessary motion, and limits the manipulator's ability to utilize the redundancy for other important tasks such as obstacle avoidance.
3) Since only the projection of gradient onto the null space is used to determine the amount of self-motion, GPM optimization ignores the information of the gradient component along the row space of the Jacobian. If the starting configuration is close to a local maximum when the performance criterion is to be minimized and vice-versa, the projection $(I - J^+J)\nabla H$ is very small. Thus, the amount of self motion utilized to avoid the joint limits is very small. The resulting joint trajectory is very close to the LN trajectory. If using the LN solution leads the joint to its limit in this case, the joint limit may not be avoided by GPM.

V. Weighted Least-Norm Solution to Avoid Joint Limits

The WLN solution for (1) is shown in (9) for the weighted norm defined by (3). We assume the weighting matrix W in (3) to be a diagonal $n \times n$ matrix with the following form:

$$W = \begin{bmatrix} w_1 & 0 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & 0 & \cdots & w_n \end{bmatrix} \tag{13}$$

where w_i, the ith element of the diagonal matrix W, is defined as

$$w_i = 1 + \left| \frac{\partial H(\underline{\theta})}{\partial \theta_i} \right| \tag{14}$$

and $H(\underline{\theta})$ is the performance criterion defined in (12) for avoiding joint limits. From (12), we can show the following:

$$\frac{\partial H(\underline{\theta})}{\partial \theta_i} = \frac{(\theta_{i,\max} - \theta_{i,\min})^2 (2\theta_i - \theta_{i,\max} - \theta_{i,\min})}{4(\theta_{i,\max} - \theta_i)^2 (\theta_i - \theta_{i,\min})^2}. \tag{15}$$

It may be noted that $\partial H/\partial\theta_i$ is equal to zero if the ith joint is at the middle of its range, and goes to infinity at either limit. Thus it is evident from the definition of w_i, that its value goes to infinity at the joint limits and is equal to 1 if the ith joint is at the middle of its range. Therefore, if a joint gets close to its limit, its weighting factor gets high, and it causes reduction in its motion. Since very close to the joint limit the weighting factor is near infinity, the corresponding joint virtually stops and joint limit avoidance is guaranteed.

The above WLN method as it is, does not differentiate whether the joint is moving toward or away from the limits. If the joint is moving away from the limit, there is no need to penalize its motion even though the $|\partial H/\partial\theta_i|$ value is high. In such a case, letting the joint move freely will allow the redundancy to be useful for other purposes such as avoiding obstacles. Taking this into account, we redefine the weighting factors as follows:

$$w_i = \begin{cases} 1 + \left|\frac{\partial H(\underline{\theta})}{\partial \theta_i}\right| & \text{if } \Delta\left|\frac{\partial H}{\partial \theta_i}\right| \geq 0 \\ 1 & \text{if } \Delta\left|\frac{\partial H}{\partial \theta_i}\right| < 0. \end{cases} \tag{16}$$

As shown above, w_i is not a continuous function of joint angles. It may be discontinuous when $\Delta|\partial H/\partial\theta_i|$ changes sign. Notice, when a joint moves toward its limit, the value of $\Delta|\partial H/\partial\theta_i|$ increases. This value is zero when joint velocity is zero, and it is negative when the joint is moving away from its limit. $\Delta|\partial H/\partial\theta_i|$ changes sign either in the middle of the corresponding joint range where it is zero itself and joint velocity is not necessarily zero, or away from the middle when the corresponding joint velocity changes sign and is zero in value. Thus, we have the following two possible cases:

1) In the middle of the joint range, where from (16) we obtain $w_i = 1$ for $\Delta|\partial H/\partial\theta_i| >= 0$ as well as $\Delta|\partial H/\partial\theta_i| < 0$. Thus there is not discontinuity in w_i.
2) Away from the middle of the corresponding joint range, where w_i changes from a large positive value to 1, or from 1 to a greater than 1 value when $\Delta|\partial H/\partial\theta_i|$ changes sign. Since the corresponding joint velocity is zero at these points, it does not affect its continuity.

Thus defining w_i to be discontinuous in (16), does not affect the continuity of the joint velocities.

If the continuity of joint acceleration is important for a given task, then we can specify the function of w_i for smooth transition between the cases of $\Delta|\partial H/\partial\theta_i| >= 0$ and $\Delta|\partial H/\partial\theta_i| < 0$. Since weighting matrix W is defined as a diagonal matrix in (13), W^{-1} is a diagonal matrix with entries as reciprocals of the corresponding entries of W. Therefore, the overall calculations involved in WLN solution defined by (9) are less than the calculations involved in GPM defined by (10), which requires the computation of the homogeneous solution in addition to the LN solution.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.

VI. Comparison of Weighted Least-Norm Solution Scheme with the Gradient Projection Method

The WLN based scheme was implemented in real-time to control the 7-DOF Robotics Research Corporation manipulator (Fig. 1). The method proposed in [14] was used to obtain the least-norm solution of (5) for efficient calculations. The algorithm was implement on a Silicon Graphics Workstation with R3000 CPU and a floating point processor running at 33 Mhz clock. It takes about 6×10^{-4} seconds to obtain the WLN solution, which is fast enough to meet the RRC servo requirement of 2.5×10^{-3} seconds. This system can run in three different modes: graphics simulation mode, real-time manipulator control mode, and simultaneous graphics and real-time control mode. The GPM and the LN solutions were implemented for comparison.

It may be noted that WLN and GPM are two different control schemes even though they may use the same cost function for optimization. From (10) we see that $\dot{\underline{\theta}}_{SG}$, the self-motion for GPM, is:

$$\dot{\underline{\theta}}_{SG} = k(I - J^{+}J)\nabla H. \tag{17}$$

By subtacting the LN solution from the WLN solution, we can obtain $\dot{\underline{\theta}}_{SW}$, the self-motion for WLN as:

$$\dot{\underline{\theta}}_{SW} = (W^{-1/2}J_W^{+} - J^{+})\dot{\underline{x}}. \tag{18}$$

From (17), we see that $\dot{\underline{\theta}}_{SG}$, the self-motion used by GPM, depends only on the manipulator configuration. However, $\dot{\underline{\theta}}_{SW}$, the self-motion used by WLN, depends not only on the configuration, but also on the end-effector velocity vector $\dot{\underline{x}}$. Therefore, the magnitudes and directions of $\dot{\underline{\theta}}_{SG}$ and $\dot{\underline{\theta}}_{SW}$ are different in general for a manipulator with multiple degrees of redundancy. For a manipulator with one degree of redundancy, $\dot{\underline{\theta}}_{SG}$ and $\dot{\underline{\theta}}_{SW}$ are in the same direction. However, the magnitudes of $\dot{\underline{\theta}}_{SG}$ & $\dot{\underline{\theta}}_{SW}$ are different. While the magnitude of $\dot{\underline{\theta}}_{SG}$ is fixed for a given configuration, $\dot{\underline{\theta}}_{SW}$ magnitude changes according to the direction and the magnitude of the end-effector velocity vector $\dot{\underline{x}}$. Therefore, the two schemes should produce different joint trajectories for the same end-effector path using the same cost function. When the end-effector is not moving, WLN scheme does not change its configuration, but GPM algorithm continues to change its configuration until it reaches an optimum.

For better comparison, a constant end-effector speed is used for LN, WLN, and GPM control schemes in the following examples that show the advantages of the WLN solution.

A. Example 1

Figs. 4 and 5 show the 3-D maps corresponding to the straight line end-effector trajectory shown in Fig. 5. The performance criterion $H(\underline{\theta})$ defined in (12), is plotted as a function of the end-effector trajectory and self-motion. Since a higher value of this performance criterion represents proximity to the joint limits, we refer to it as "cost" which needs to be minimized. The "end-effector trajectory" axis represents the moving distance of the end-effector. The "self-motion" axis indicates the self motion of the manipulator. On these maps, we use joint 1 value for self-motion since it has a one-to-one correspondence with the self-motion along the end-effector trajectory.

The curves along the end-effector trajectory direction on the 3-D map in Fig. 4 are referred to as "cost curves." Separate cost curves are obtained for different initial configurations. The cost curves in Fig. 4 are for the joint trajectories obtained using the GPM to minimize the "cost." The same scalar constant "k" is used in the second part of (10) to compute these joint trajectories. It may be noted that for the starting configurations close to the "local maximum" (peak in the middle), the curves change the direction to the "lower cost area" much later compared to the curves starting away from the local maximum.

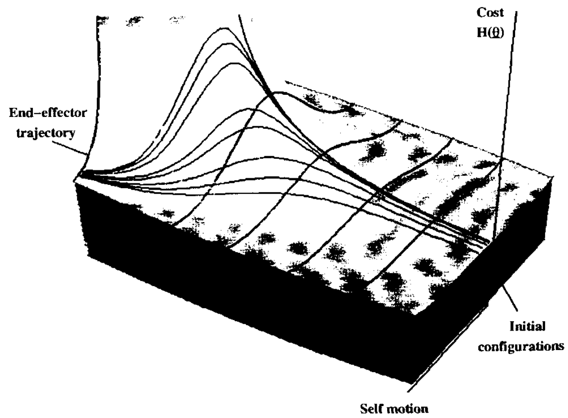


Fig. 4. 3-D cost map and GP method cost curves.

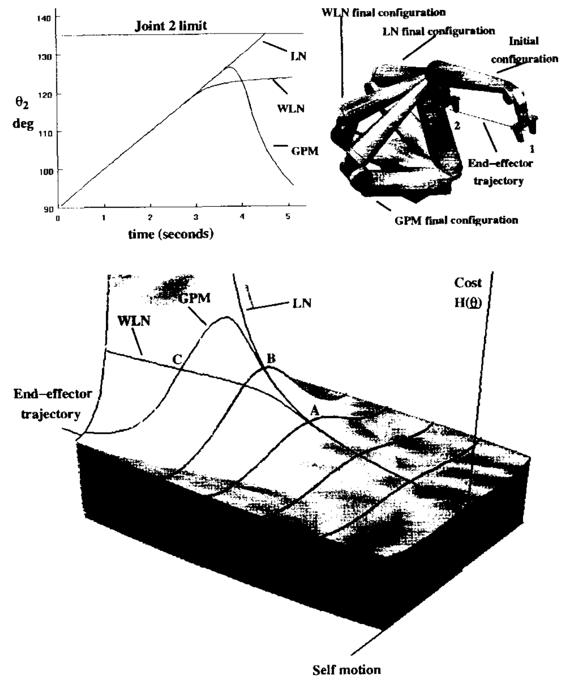


Fig. 5. Robot trajectory, joint 2 plot, and cost curves—Example 1.

If the starting configuration is too close to the local maximum, the curves do not deviate much from the LN trajectory. These curves continue to climb toward the high-cost area and hit the joint limit. It may be noted that the projection of the gradient onto the null space is zero at the maximum or minimum. Since it is very small near the local maximum, it does not produce enough self-motion to change the direction toward a low cost area. Therefore, the GPM using (10) provides very close to the LN solution.

To compare the proposed WLN scheme with LN and GPM solutions, we choose a starting configuration slightly away from the local maximum or the middle peak on the 3-D map in Fig. 5. The cost curves for the three schemes are shown on this 3-D map for the same starting configuration. Until Point A, WLN and GPM trajectories are similar to the LN trajectory. There after, WLN scheme changes its direction toward the low cost area. Since, the magnitude of the self

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.

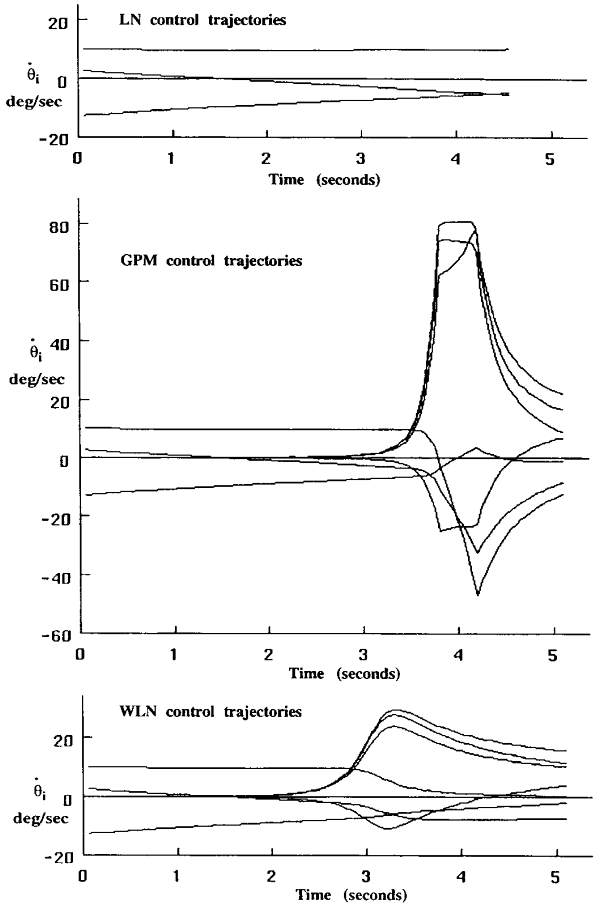


Fig. 6. Joint velocity vector $\dot{\underline{\theta}}$ components—Example 1.

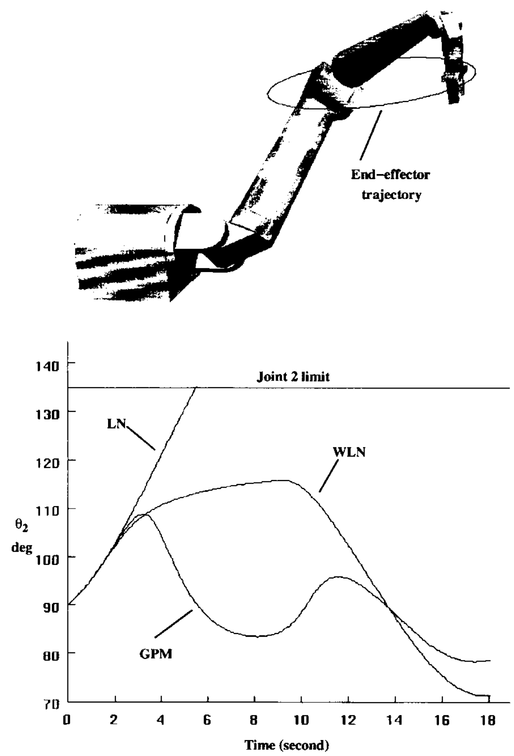


Fig. 7. Robot trajectory and joint 2 plot—Example 2.

motion in the WLN scheme is affected by the overall magnitude of the gradient (and not just on the projection in the null space direction), the deviation from the LN path is significant in this case. On the other hand, as the projection of the gradient onto the null space is too small, the GPM trajectory in this region is similar to the LN trajectory. Thus during the trajectory between A and B, the magnitude of the self-motion for the GPM trajectory is much smaller compared to the WLN trajectory. After point B, however, the change in the performance criterion of the gradient along the null space is larger. The GPM begins to work at this stage and results in a trajectory away from the LN trajectory and toward the low cost area.

At point C we note another interesting phenomena. Here both the WLN and GPM trajectories reach the same configuration. Since the projection of the gradient along the null space is relatively high at this point, the GPM uses a large self-motion to move toward a low-cost area. However, the overall gradient at this point is not much higher than before; and thus, the self-motion in WLN solution does not increase much. Here, unlike the GPM, the WLN solution does not attempt to reach the minimum since it is away from the joint limit and is in the safe region. It may be noted that if the problem of the GPM responding too late is corrected between A and B by using a higher value of gain k, the self-motion will get too large in the region toward C since the projection of the gradient along the null space is already too large. On the other hand, a low-gain k to reduce the self-motion after C will make the situation undesirable between B and C. In that case, the trajectory will not deviate much from the LN trajectory, and it may be too late to avoid the joint limits. Thus it is not possible to find a k which would work well throughout the trajectory. After point A, the cost function of WLN is the same as GPM since none of the joints changes the direction of motion, and the desired end-effector trajectory requires the joints to move toward their limits. However, as seen in Fig. 5, the two joint trajectories are quite different.

Fig. 5 also shows the joint 2 angle plots for the three schemes. As shown LN runs into the joint limit. WLN slows down the speed of joint 2 and stops it from moving toward its limit, and GPM moves joint 2 away from its limit. The GPM effect on the trajectory is much later and more severe compared to the WLN scheme.

The joint velocity plots for the three schemes are shown in Fig. 6. Since the projection of the gradient onto the null space is too large during the time $t = 3.3$ to 4 seconds, the joint velocities go over their limits for the GPM trajectory. To restrict the joint velocities within the hardware bounds, the magnitude of the self-motion is reduced by reducing k [13]. The total joint motion cost $\int \Sigma\dot{\theta}^2\, dt$ is equal to 4.09 in this case. It is apparent from this plot that the WLN scheme does not result in unnecessary self-motion or elbow motion, and avoids joint limits at the same time. The maximum joint velocities are much lower in this case as compared to GPM. The total motion cost $\int \Sigma\dot{\theta}^2\, dt$ is equal to 1.04 for the WLN method. It is only 25% of the motion cost for GPM. The maximum joint velocities for LN are the lowest of all the three schemes. However, it results in joint 2 going over its limit.

B. Example 2

In this example we choose a circular trajectory for the end-effector as shown in Fig. 7. Also shown in Fig. 7 are the joint 2 trajectories

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.

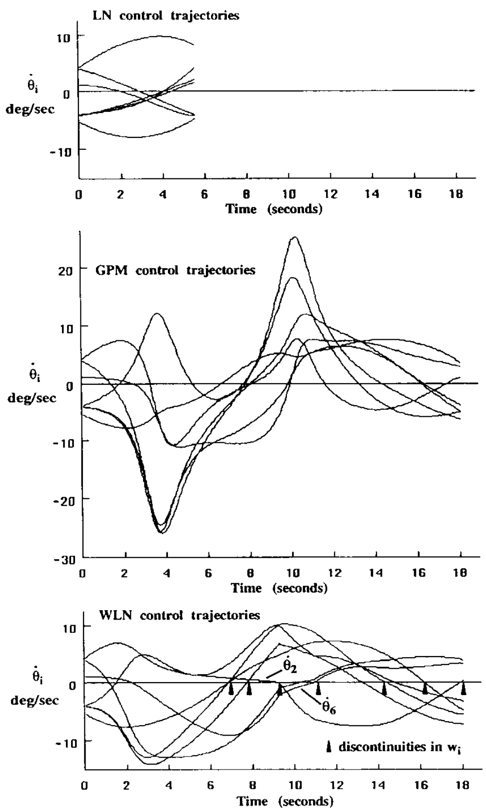


Fig. 8. Joint velocity vector $\dot{\underline{\theta}}$ components—Example 2.

for the LN, GPM, and WLN schemes. Fig. 8 shows the plots of joint velocities for the three schemes. As in the previous example, the joint velocities resulting from the GPM are much higher than the WLN control scheme. The total motion cost of the GPM is $\int \Sigma\dot{\theta}^2\, dt = 2.6$. When using the WLN, the total cost is $\int \Sigma\dot{\theta}^2\, dt = 1.29$, which is only 50% of the GPM. In Fig. 8, there are 7 points where w_i, the weighting coefficients of the WLN are not continuous. As discussed earlier, at all the points where the weighting coefficients are discontinuous, the corresponding joint velocities are zero. Only at two of these points the acceleration is discontinuous. One of these is at $t = 9.2$, where the weight of joint 2 changes from 8.34 to 1 because joint 2 has to move away from its limit to follow the end-effector trajectory. Since $\dot{\theta}_2$ is zero at this point, the sudden change in the weighting coefficient does not cause a discontinuity in the joint 2 velocity. Another point is at $t = 11$, where joint 6 moves away from its limit and causes w_6 to change from 3.146 to 1. Again $\dot{\theta}_6$ is zero at this point and thus the joint velocity is continuous.

When the above WLN and GPM trajectories were computed in real-time and used to run the RRC K-2107 manipulator, the WLN trajectory motion appeared smother and less noisy compared to the GPM trajectory. This is in spite of the discontinuity in the acceleration at two points due to the discontinuity in the weighting factors. The higher noise and vibration in the GPM trajectory appear to be due to higher velocities and accelerations.

Repeating the same end-effector trajectory a few times, both GPM and WLN trajectories become repeatable. As shown in Fig. 8, LN control scheme reaches a joint limit before completing the trajectory.

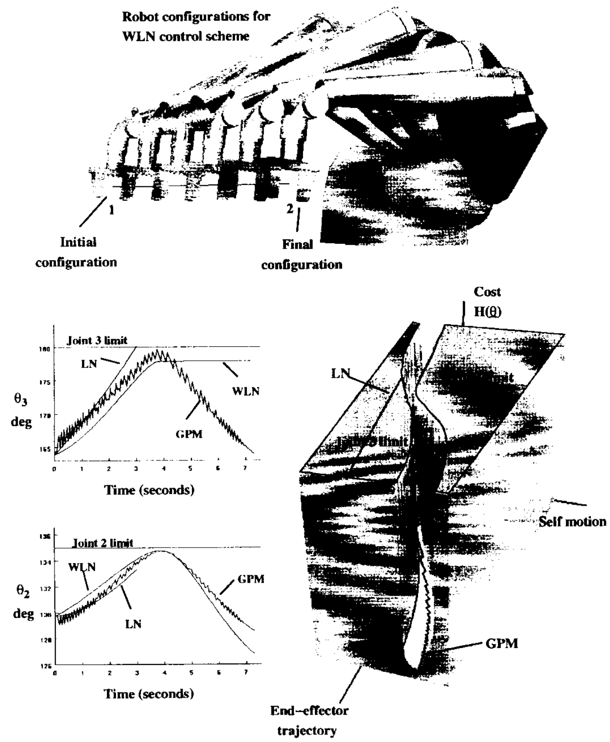


Fig. 9. Robot trajectory, joints 2 and 3 plots, and cost curves—Example 3.

C. Example 3

The end-effector trajectory for this example is shown in Fig. 9. The usable workspace in this case is restricted by the limits on joints 2 and 3. Fig. 9 also shows the 3-D map with cost curves and plots of joints 2 and 3 for the LN, GPM, and WLN trajectories. From the 3-D map we see that the available workspace is in the "valley" of the cost map. The cost at the limits of joints 2 and 3 is equal to infinity, and is shown by sectional lines on the 3-D map. In this example also, the LN method reaches the joint limit before completing the trajectory. Since the two joint limits are so close to each other, using self-motion to avoid one joint limit leads to the other limit. The projection of the gradient onto the null space is extremely large even when the manipulator is not far from the optimum configuration. Thus, GPM results in extremely high joint velocities and oscillations about the optimum configuration. Even though GPM avoids joint limits in this example, the oscillations in the joint velocities, as shown in Fig. 10, are unacceptable. Even the smallest value of "k" that avoids joint limits, causes very large oscillations. The results shown in Fig. 10 are obtained from simulations as the real-time control is likely to damage the manipulator in this mode. WLN works very well in this example even though the weights were not continuous at some points. As shown in Fig. 10, the joint velocities do not oscillate in this case, and are much smaller in magnitude. Of course, the oscillation about the optimum using GPM can be reduced by reducing the sampling step size. However, for the same sampling step size, which is limited by the computer hardware, WLN scheme performs much better than GPM algorithm as shown in this example.

VII. Summary and Conclusions

A new scheme using the WLN solution was introduced for avoiding joint limits. It was implemented in real-time for the control of a 7-

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.

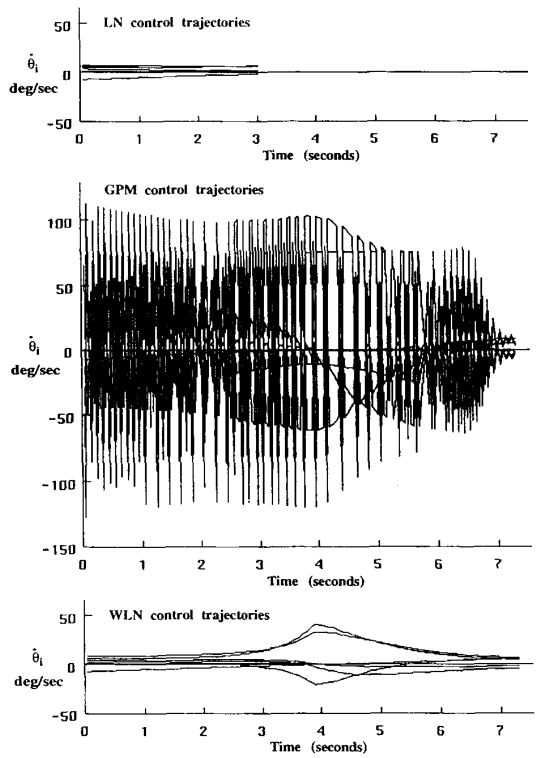


Fig. 10. Joint velocity vector $\dot{\underline{\theta}}$components—Example 3.

DOF RRC manipulator. Using three different end-effector trajectories of the RRC manipulator, a comparison was made with the GPM for avoiding joint limits.

The main problem with the GPM is in the selection of the magnitude of the self-motion. It is the product of the scalar constant k and $(I - J^+J)\nabla H$. If, for example, a performance criterion is to be minimized and the manipulator is close to a local maximum, then $(I - J^+J)\nabla H$ term is very small, resulting in little self-motion to move the manipulator toward a minimum. If this problem is solved by increasing k, then it creates unnecessarily high self-motion when $(I - J^+J)\nabla H$ gets larger in value. A large k also results in oscillations near a minimum.

The WLN solution takes care of this problem of selecting the right magnitude of self-motion. While it dampens any joint motion in the direction of joint limits, it does not try to actively maximize the distance of the joints from their limits. Thus, it avoids unnecessary self-motion and oscillations. Whenever the gradient of the performance criterion with respect to a particular joint is high indicating proximity to the limit of that joint, its motion is penalized by adding appropriate self motion. This is unlike the GPM, where only the projection of the gradient along the null space is used to determine the magnitude of the self motion.

From the three examples used to compare WLN and GPM trajectories, the following observations were made:

1) The WLN scheme uses self-motion only when it is necessary, thus resulting in lower joint motion cost defined by $\int \Sigma\dot{\theta}^2\,dt$ compared to GPM. As discussed before, this difference is because while WLN only dampens the motion in the direction of joint limits, GPM actively utilize the self motion to maximize the distance from joint limits.
2) The WLN method guarantees the joint limit avoidance unlike the GPM. This is because complete gradient, and not just its component along the null space, is used to determine the magnitude of the self motion.
3) Unlike GPM, the WLN joint trajectories are oscillation-free in general for a fixed sampling rate. Specifically, if the cost curve has to go through a narrow valley on the 3-D map to avoid joint limits, it is very difficult to choose the k value that works with GPM and avoids oscillations. However, the WLN method worked very well for such a trajectory and there were no oscillations.
4) When the manipulator moves away from the joint limit while following the end-effector trajectory, the proposed WLN-based scheme utilizes the redundancy solely for the purpose of tracking the end-effector trajectory or, if needed, for optimizing any other criterion. It does not unnecessarily waste the self-motion for avoiding joint limits in this case. Such a switch does not introduce any discontinuities in the joint velocities.

We are currently looking into using the WLN scheme for utilizing redundancy for avoiding obstacles. As in the case of joint limits, it is important to avoid the obstacles, but irrelevant to maximize the distance from the obstacles.

References

[1] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *ASME J. Dynamic Syst., Measure., and Cont.*, vol. 94, no. 4, 303–309, 1972.

[2] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 12, 1977.

[3] C. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 3, pp. 245–250, Mar. 1983.

[4] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1985, pp. 1016–1021.

[5] K. C. Suh and J. M. Hollerbach, "Local versus global optimization of redundant manipulators," in *Proc. 1987 IEEE Int. Conf. on Robot. and Automat.*, 1987, pp. 619–624.

[6] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *The Int. J. Robot. Res.*, vol. 6, no. 1, pp. 32–42, 1987.

[7] R. V. Dubey and J. Y. S. Luh, "Redundant robot control for higher flexibility," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1987, pp. 1066–1072.

[8] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics Research: The First International Symposium*, M. Brady and R. Paul, Eds. Cambridge, MA: MIT Press, 1984, pp. 735–747.

[9] J. Angeles and O. Ma, "Quadmin: An integrated package for constrained nonlinear least-square problems," in *Proc. 1989 ASME Int. Comput. Eng. Conf.*, vol. 2, pp. 247–252.

[10] F.-T. Cheng, T.-H. Chen, and Y.-Y. Sun, "Efficient algorithm for resolving manipulator redundancy—The compact QP method," in *Proc. 1992 IEEE Int. Conf. Robotics Automat.*, 1992, pp. 508–513.

[11] J. A. Euler, R. V. Dubey, S. M. Babcock, and W. R. Hamel, "A comparison of two real-time control schemes for redundant manipulators with bounded joint velocities," in *Proc. of IEEE Int. Conf. Robot. and Automat.*, 1989, pp. 106–112.

[12] A. Albert, *Regression and the Moore-Penrose Pseudo-Inverse.* New York: Academic Press, 1972.

[13] J. A. Euler, and R. V. Dubey, and S. M. Babcock, "Self-motion determination based on actuator velocity bounds for redundant manipulators," *J. Robot. Syst.*, vol. 6, no. 4, pp. 417–425, Aug. 1989.

[14] R. V. Dubey, J. A. Euler, and S. M. Babcock, "Real-time implementation of an optimization scheme for seven-degree-of-freedom redundant manipulators," *IEEE Trans. Robot. and Automat.*, vol. 7, Oct. 1991.

[15] H. Zghal, R. V. Dubey, and J. A. Euler, "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1990, vol. 2, pp. 1006–1011.

Authorized licensed use limited to: KTH Royal Institute of Technology. Downloaded on February 21,2020 at 15:15:18 UTC from IEEE Xplore. Restrictions apply.