A Varied Weights Method for the Kinematic Control of Redundant Manipulators with Multiple Constraints

Ji Xiang Member, IEEE, Congwei Zhong, and Wei Wei

Abstract

A varied weights (VW) method has been proposed in this paper for the kinematic control of redundant manipulators with multiple constraints. A weight factor rule to reflect the transition of the state of constraint subtask between activeness and inactiveness is presented. Each constraint has a varied weight factor in such a way that every time only the main task and the active constraint subtasks are considered. A new concept of effective singular value is presented for the design of damping factors to avoid the pseudo-singularity arising from the transition of weight factors. The experiments on the seven-degree-of-freedom Ping-Pong manipulator illustrate the efficacy of the proposed VW method.

Index Terms

Varied weights (VW) method, kinematic control, redundant manipulator, multiple constraints, effective singular value.

I. INTRODUCTION

Manipulators that are able to move like the human arm have always been one of the long-term goals in the field of robotics. One critical problem for this goal is how to control the motion of manipulators in the dynamical environment where the environment might have a unpredictable change and the tasks probably vary as the requirement changes. In general, the control law of manipulators can be designed in either the joint space or the task space. To control

This research was supported by a grant from the National 863 Program of China (2011AA050204), the National Natural Science Foundation of China (60704030, 61074122, 61104149) and the Fundamental Research Funds for the Central Universities (2011QNA4010).

Ji Xiang and Wei Wei are with the department of System Science and Engineering, College of Electrical Engineering, Zhejiang University, P. R. China.

Ji Xiang is the corresponding author. Email: jxiang@zju.edu.cn.

Congwei Zhong Li is with the China Ship Development and Design Center, Shanghai, P. R. China.
manipulators in the joint space, both the motion replanning (for dynamical environments) in the task space and the inverse kinematics solution are necessary, which are too hard and complex to compute on-line [1]–[3]. So for the dynamical environments, it had better design the control law in the task space.

Control of manipulators in the task space is often subjected to singularity, joint limits and obstacles, among other things. To improve the ability to overcome these problems in the dynamical environment, redundant manipulators have often been adopted with the penalty of increased hardware costs and control complexities. A redundant manipulator has more degree of freedom (DOF) than those strictly required by the given task. The additional DOFs lead to the characteristics of self-motion that manipulators move with the pose of end-effector unchanged. Exploring the self-motion, the redundant manipulator can execute some subtasks with the main task guaranteed firstly.

There are many approaches reported in literature for the control of redundant manipulators with additional subtasks. The prevalent one is the gradient project method (GPM), which was first presented in [4] for the avoidance of joint limits. There the self-motion is designed to move along the gradient of a quadratic cost function of joint limits which reaches the minimal value when all joints are at the middle position of their feasible regions. This idea has been widely used in lots of works for various subtasks, including avoidances of singularity [5] [6], minimization of joint torque energy [7] and visual servorobots [8]. Both the extended Jacobian method (EJM) [9], [10] and the augment Jacobian method (AJM) [11], [12] take the additional subtasks as the part of main tasks so that the Jacobian matrix becomes a square one. The major advantage of EJM and AJM is repeatability, that is, the joint-space trajectory corresponding to a cyclic task-space trajectory is cyclic. The weighted least norm (WLN) method was presented in [13] for the avoidance of joint limits of redundant manipulators. There the Jacobian matrix is weighted by a varied matrix dependent on the distances away from the joint limits. The WLN method is more efficient for the avoidances of joint limits than the other methods mentioned above in that it preplans no motion but forbids any motion to violate joint limits.

In a complex environment, there are often many constraints to be involved in the motion of manipulator. For example, in the environment for playing Ping-Pong, the collision avoidance for the table surface and robot body are necessary to consider, beside the joint limits. The control
of manipulator under multiple constraints is difficult and most reported methods can no longer work well. The WLN method is effective only for the joint limits. The GPM method can be nestedly utilized for the case of multiple subtasks [14], [15], where all subtasks are assigned different priorities and the subtask of low priority is projected into the null space of last subtask. The nested GPM method can not reflect the fact that the priorities of constraints are varied as the configuration of manipulator varies, and moreover it is known that the GPM method locally optimizes the subtasks only in the null space of main tasks such that the constraint tasks can not often be ensured by the GPM method [16]. Cleary and Tesar [17] weighted each subtask and summed them as an overall performance criterion, and then the GPM method were used to deal with multiple subtasks. There the weight factors of subtask are constant. Shen and Gu [18] presented a solution to optimize the weighted-norm sum of all tasks including the main task and the subtasks, but this method could lead to the poor performance for all tasks. More recently, the general weighted least norm (GWLN) method was proposed to cope with the case of multiple constraints in [19]. There each constraint is formulated by a virtual joint with unilateral joint limit. Inheriting the good performance of WLN method for avoiding joint limits, the constraint tasks can be effectively guaranteed by the GWLN method applied in the virtual joint space. The GWLN method can cope with many constraint tasks by combining the constraints that would not happen simultaneously, into one subtask. However, it is often difficult to judge whether a pair of constraints can not simultaneously happen, especially when the main task is varied, e.g., playing Ping-Pong.

In this paper, we will present a variable weights (VW) method to cope with the case of multiple constraints. It should be recognized that the idea of variable weights has been extensively utilized in many fields. Particularly there are several works on kinematical control of redundant manipulators with multiple criteria. In [20] and [21], two different weight functions were respectively presented for the subtasks. There the sum of the weighted criteria of subtasks is projected on the null space of Jacobian matrix to guarantee the main task firstly. In [22], the sum of the weighted criteria of subtasks was optimized in the space spanned by the submatrix of Jacobian matrix. More closely related work is the method proposed in [23], where the control law is very similar that of the nested GPM method with the Jacobian matrix of each hierarchy being binary weighted. An original inverse-operator method to ensure the continuity of pseudoinverse of weighted Jacobian matrix. However, as the number of considered tasks increases, the inverse-operator method seems very complicated and requires a lot of computing time. Different form
them, the presented (VW) method gives equal treatments to all tasks with only difference being that the constraint tasks have variable weight factors. Each constraint is weighted by a varied factor that determines the constraint priority in the current configuration. When necessary, the constraint task will have the higher priority than that of the main task so that the constraint task will be guaranteed firstly; when unnecessary, the constraints will have zero weight factor so that the main task will be executed without any influence of the constraints. In such a formulation of VW method, there are at least two challenging problems to consider: one is how to design a weight factor rule to reflect the transition between the inactive state and the active state of constraint without causing a jerk of control commands; the other is how to cope with the pseudo-singularity due to the weights transition so as to guarantee the performance of main task as soon as possible.

The remainder of this paper is organized as follows. Problem is formulated in Section II. The VW method is presented in Section III including four subsections. The weighted optimal problem and its prototype solutions are addressed in the first subsection, followed by the algorithm of VW method given by the second subsection. The rule of weight factor and the design of damped factor are presented in the third subsection. The performance of VW method is analyzed in the last subsection. The experiments on 7-DOFs Ping-Pong manipulator under the proposed VW method are reported in Section IV. Section V concludes the paper.

II. Problem Formulation

A. Kinematic control and basic solutions

Let \( \dot{q} \in \mathbb{R}^n \) be the vector to represent the joint velocities and \( \dot{x} \in \mathbb{R}^m \) the vector to represent the velocity of end-effector in the task space. The kinematic relation between them is

\[
\dot{x} = J(q)\dot{q}
\]

where \( J(q) \) is the \((m \times n)\) configuration dependent Jacobian matrix. The kinematic control considered in this paper is to obtain the joint velocity from the above relation such that \( \dot{x} = \dot{x}_d \) where \( \dot{x}_d \) represents the desired velocity of end-effector for the main task. For nonredundant manipulators, \( m = n \) and the solution of (II.1) is unique whenever the Jacobian matrix is of full rank; while for redundant manipulators, \( m < n \) and there are infinite solutions \( \dot{q} \) satisfying equation (II.1). If \( J(q) \) is of full row rank, the well-known pseudoinverse gives the following solution [24]

\[
\dot{q}_I = J^+\dot{x}_d = J^T(JJ^T)^{-1}\dot{x}_d,
\]
where $J^+$ is the pseudo-inverse of the Jacobian matrix. This solution just solves the following Least-squares minimum-norm problem (LMP):

$$\text{LMP : } \min \| \dot{q} \| \text{ subject to: } \min \| \dot{x}_d - J \dot{q} \|, \quad (\text{II.3})$$

which means that the minimal effort ($\min \| \dot{q} \|$) in the level of velocity is used to realize the main task, namely, $\dot{x}_d - J \dot{q} = 0$. In this sense, control law (II.2) is preferable if the Jacobian matrix has full row rank.

However, there always are some configurations at which $J(q)$ is either rank deficient or ill-conditioned. In such configurations, high joint velocities will be required even for very small $\dot{x}_d$ in the restricted or singular directions. To overcome this drawback, one well-known way is the damped least square method [25],

$$\dot{q}_{il} = (J^T J + \lambda^2 I_n)^{-1} J^T \dot{x}_d \quad (\text{II.4})$$

where $I_n$ is the identity matrix of $n$ dimensions, and $\lambda > 0$ is the damping factor. Formula (II.4) is nothing but the solution of the following damped Least-squares problem (DLP):

$$\text{DLP: } \min \| \dot{x}_d - J \dot{q} \|^2 + \lambda^2 \| \dot{q} \|^2, \quad (\text{II.5})$$

which means that the control law (II.4) is a compromise product between the requirement of accuracy of the main task and the feasibility of the joint velocities. The extent of compromising determined by the damping factor $\lambda$: the larger value of $\lambda$ implies smaller joint velocities norm but the bigger deviation of the main task.

Notice that for redundant manipulators, $\lambda$ must be nonzero for the feasibility of (II.4) because $J^T J$ will never be of full rank for redundant manipulators. However, by the singular value decomposition (SVD) of $J$ it is easy to rewrite (II.4) as

$$\dot{q}_{il} = J^T (JJ^T + \lambda^2 I_m)^{-1} \dot{x}_d, \quad (\text{II.6})$$

which permits the zero damping factor provided that $J$ is of full row rank. When $\lambda = 0$, formula (II.6) reduces to (II.2), the solution of LMP. In fact (II.6) instead of (II.4) has been extensively utilized to effectively cope with the problem of ill-conditioned or singular Jacobian.

Both (II.2) and (II.6) are the basic kinematic control laws for redundant manipulators. The prevalent strategy is to design a varied damping factor $\lambda$ such that the control law, denoted by $\dot{q}_c$, smoothly switches between (II.2) and (II.6) such that nonzero $\lambda$ only happens whenever necessary.
B. Multiple Constraints

In real applications, there are often many constraints, such as the joint limitations and the finite torque of joint, to influence the performance, even to prevent the realization of the basic control law, (II.2) or (II.6) in the last subsection. In this paper, we assume that the constraints from which the manipulators suffer are unilateral and can be represented in the style of inequality, namely,

\[ f_i(q) > 0, \quad i = 1, \cdots, r, \quad \text{(II.7)} \]

where \( f_i(\cdot) : \mathbb{R}^n \mapsto \mathbb{R} \) denotes the function of \( i \)th constraint and \( r \) is the total number of constraints. For example, the joint limitation of first joint can be represented by two constraints,

\[
\begin{aligned}
  f_1 &= q_1 - q_{1\text{min}} > 0 \\
  f_2 &= q_{1\text{max}} - q_1 > 0 
\end{aligned}
\quad \text{(II.8)}
\]

or by one constraint,

\[ f_1 = (q_{1\text{max}} - q_1)(q_1 - q_{1\text{min}}) > 0. \quad \text{(II.9)} \]

where \( q_1, q_{1\text{min}} \) and \( q_{1\text{max}} \) are the position, low bound and upper bound of the first joint, respectively.

We hope that when \( f_i(q) \) is far away from 0 the manipulator moves by itself without considering constraints; while \( f_i(q) \to 0 \), simply to stop \( f_i(q) \) to guarantee \( f_i(q) > 0 \). In this sense, constraint (II.9) in the level of position can be further explained in the level of velocity,

\[ J_{f_i}(q)\dot{q} = 0, \quad \text{if } f_i(q) \leq \epsilon_{f_i} \quad \text{(II.10)} \]

where \( J_{f_i}(q) \) is the Jacobian matrix of \( f_i(q) \), and \( \epsilon_{f_i} \) is the threshold denoting the dangerous region of the \( i \)th constraint.

In this way, we can uniformly formulate the kinematic control of manipulators with \( r \) constraints in the term of multiple tasks as follows

\[
\begin{aligned}
  J(q)\dot{q} &= \dot{x}_d \\
  J_{f_i}(q)\dot{q} &= b_{f_i}, \quad i = 1, \cdots, r, 
\end{aligned}
\quad \text{(II.11)}
\]

where \( b_{f_i} \) is a designed parameter reflecting the constraint requirement. For formula (II.10), one has

\[ b_{f_i} = \begin{cases} 
  0 & \text{if } f_i(q) \leq \epsilon_{f_i} \\
  \text{arbitrary scalar} & \text{or else.} 
\end{cases} \quad \text{(II.12)} \]
Let \( A = [J^T, J_{f1}^T, \ldots, J_{fr}^T] \) and \( b = [\dot{x}_d^T, b_{f1}^T, \ldots, b_{fr}^T] \), then (II.11) has the following compact form of
\[
A \dot{\mathbf{q}} = b. \tag{II.13}
\]

Clearly, \( A \in \mathbb{R}^{(m+r) \times n} \) and \( b \in \mathbb{R}^{m+r} \). If \( m + r \leq n \), it is the standard kinematic control problem of redundant manipulators. But when \( m + r > n \), (II.13) is an overdetermined equation. Throughout this paper, it is assumed that \( m + r > n \). This means that the number of considered constraints is larger than the redundant DOFs of manipulators. If \( A \) has full column rank, the approximate solution of (II.13) to minimize \( \|A \dot{\mathbf{q}} - b\| \) is,
\[
\dot{\mathbf{q}} = (A^T A)^{-1} A^T b. \tag{II.14}
\]

This solution is not good due to paying the identical attention to different tasks. In fact it is the main task that we want to realize in the whole process, while the constraint task is ensured only when necessary, namely, \( f_i(q) < \epsilon_i \). If possible, we want to on-line adjust the equations of (II.13) such that anytime it only consists of the main task and the active constraint tasks.

Motivated by this consideration, this paper presents a VW method to solve the problem formulated by (II.11). The constraint tasks will have time-varying weights dependent on the value of \( f_i(q) \) in such a way that the weight of the \( i \)th constraint is very large if \( f_i(q) \leq \epsilon_i \), or else it is near to zero. Our goal is to achieve the performance that, *when the constraints are far away from being activated, try the best to realize the main task; while the constraints are active, still try the best to realize the main task after the active constraints are guaranteed firstly.* This performance might be the optimal one in the case where constraints exist.

### III. Varied Weight Method

#### A. Prototype solutions

Let \( w_i > 0 \) be the weight factor associated with the \( i \)th constraint task. Consider the following weighted optimal problem,
\[
\min \|J(q) \dot{\mathbf{q}} - \dot{x}_d\| + \sum_{i=1}^{r} w_i^2 \|J_{f_i}(q) \dot{\mathbf{q}} - b_{f_i}\| + \lambda^2 \|\dot{\mathbf{q}}\|. \tag{III.1}
\]

Define the following matrices
\[
W = \begin{bmatrix} w_1 & \cdots & w_r \end{bmatrix}, \quad \bar{W} = \begin{bmatrix} I_m & W^2 \end{bmatrix}.
\]
The solution of optimal problem (III.1) is given by

\[ \dot{q}_w = (A^T \bar{W} A + \lambda^2 I_n)^{-1} A^T \bar{W} b. \]  

(III.2)

Let \( A_w = \sqrt{\bar{W}} A \) and \( b_w = \sqrt{\bar{W}} b \). Using the SVD of \( A_w \), (III.2) can be expressed as

\[ \dot{q}_w = A_w^T (A_w A_w^T + \lambda^2 I_{m+r})^{-1} b_w. \]  

(III.3)

Here the damping factor \( \lambda \) is, similar with that in DLP, to ensure the feasibility of the joint velocities with the price of derivation of the tasks.

When \( m + r > n \), \( \lambda = 0 \) is absolutely forbidden for formula (III.3). Therefore formula (III.2) seems preferable here, different from the DLP case for which solution (II.6) is preferable.

If \( b_f_i \) is defined as that in (II.10), that is \( b_f_i = 0 \), then \( b_w = [x_d^T, 0]^T \). In this case, when all constraints are far away from being activated, namely \( W = 0 \), formula (III.2) reduces to

\[ \dot{q}_w = (J_T J + \lambda^2 I_n)^{-1} J_T \dot{x}_d; \]  

(III.4)

while formula (III.3) reduces to

\[ \dot{q}_w = J_T (J J_T + \lambda^2 I_m)^{-1} \dot{x}_d. \]  

(III.5)

It is clear that \( \lambda = 0 \) is permissible for (III.5) but forbidden for (III.4). In this sense, formula (III.3) is more preferable than (III.2) in the case of \( W = 0 \).

There exists a dilemma: if one selects formula (III.2), then weighted factor \( w_i \geq 0 \) can be set arbitrarily but \( \lambda = 0 \) is forbidden in any time; while if one selects formula (III.3), then \( \lambda = 0 \) is permissible but \( W = 0 \) will be forbidden when \( \lambda = 0 \). On the other hand, the selection of damping factor according to the singular property of the involved Jacobian matrix, one prevalent way to select \( \lambda \) [26], is not suitable for formula (III.2) when the weighted matrix \( W \) is varying, as shown in the following example.

**Example 1:** Assume that \( n = 3, m = 2 \) and \( r = 2 \). \( A = [A_0^T, A_1^T, A_2^T]^T \), where \( A_0 \) is the Jacobian matrix of the main task,

\[ A_0 = \begin{bmatrix} 0.69 & 0.31 & 0.95 \\ 0.39 & -0.33 & 0.94 \end{bmatrix}, \]  

(III.6)

and \( A_1 \) and \( A_2 \) are the Jacobian matrices of constraint tasks,

\[ A_1 = \begin{bmatrix} 0.69 & 0.33 & 0.94 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.51 & -0.3 & 0.9 \end{bmatrix} \]  

(III.7)

The command vector \( b \) is

\[ b^T = \begin{bmatrix} 4 & -2 & 0 & 0 \end{bmatrix}. \]  

(III.8)
With the above configuration, consider the case that the weighted factors of constraint tasks are \( w_1 = 10^{-10} \) and \( w_2 = 10^{-10} \), respectively, that means the constraints are far away from being activated. In this case, the minimum singular value of \( A_w \) is \( 10^{-11} \), therefore, the damped factor should be used. Here \( \lambda = 0.1 \) is selected. With this, using formula (III.2) yields that control command \( \dot{q}_c \) and associated tracking error \( e_m = A_0 \dot{q}_c - \dot{x}_d \) for the main task are, respectively,

\[
\dot{q}_c = \begin{bmatrix} 3.1094 \\ 7.5587 \\ -0.6344 \end{bmatrix} \quad \text{and} \quad e_m = \begin{bmatrix} -0.1140 \\ 0.1220 \end{bmatrix}.
\]  (III.9)

On the other hand, formula (III.5) can be used since the constraint tasks are inactive. Because of the minimum singular value of \( A_0 \) is 0.4869, far away from the singularity, the damping factor in (III.5) can be set to be zero. With the zero damped factor, using (III.5) yields

\[
\dot{q}_c = \begin{bmatrix} 3.2192 \\ 7.8769 \\ -0.6980 \end{bmatrix} \quad \text{and} \quad e_m = \begin{bmatrix} -0.8 \times 10^{-15} \\ 0 \end{bmatrix}.
\]  (III.10)

Comparing (III.10) with (III.9), it can be seen that the main task can be performed well with the same level of the control strength (Here we means that \( \| \dot{q}_c \| \) only has a small difference between them).

In fact, formula (III.2) can be used with a very small damping factor \( \lambda^2 = 10^{-8} \) to obtain the same control command with that of (III.10), although the minimal singular value of \( A_w \) is \( 10^{-11} \). But with this small damping factor formula (III.2) can no longer work well when the constraints tasks become active. Let \( w_1 = 10 \) and \( w_2 = 0.1 \), and then application of (III.2) with \( \lambda^2 = 10^{-8} \) produces

\[
\dot{q}_c = \begin{bmatrix} 88.8520 \\ -37.3668 \\ -52.0643 \end{bmatrix} \quad \text{and} \quad e_m = \begin{bmatrix} -3.7369 \\ 0.0429 \end{bmatrix}.
\]  (III.11)

The small damping factor can not damp the high level of the control strength because \( A_w \) is close to the singular point, although the minimum singular value here being 0.0107 is much larger than that in the aforementioned case of \( w_1 = w_2 = 10^{-10} \).

It can be concluded that two prototype solutions (III.3) and (III.4) of the varied weight method have more and less problems so that they are not qualifiable candidates for the optimal performance claimed at the bottom of Section II. In fact, there are at least two nontrivial problems in the implementation of the VW method: the feasibility of solutions and the varied weights. The
former is closely related to the design of damping factor $\lambda$ and the latter should have the rules
satisfying the good performance.

B. Implementation algorithm of VW method

Note that the equivalence of (III.2) to (III.3) and their simplifications of (III.4) and (III.5) are
based on the tool of SVD technique. Our trick is to use the SVD tool to smoothly switch among
the four solutions of (III.2)-(III.5) so that the VW method can be implemented in the sense of
optimal performance.

Assume that matrix $A_w$ can be factored as

$$A_w = UDV^T$$

(III.12)

where $U \in \mathcal{R}^{(m+r) \times (m+r)}$ and $V \in \mathcal{R}^{n \times n}$ are the orthogonal matrices, and $D \in \mathcal{R}^{(m+r) \times n}$ has the
form of

$$D = \begin{bmatrix}
\sigma_1 \\
\sigma_2 \\
\vdots \\
\sigma_n \\
0
\end{bmatrix}$$

(III.13)

where $\sigma_i$s are the singular values of $A_w$ with the order of $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n$. Substituting (III.12)
into (III.3) yields

$$q_w = VD^T(DD^T + \lambda^2 I_{m+r})^{-1}U^Tb_w = \sum_{i=1}^{n} \frac{\sigma_i U_i^Tb_w}{\sigma_i^2 + \lambda^2} V_i$$

(III.14)

where $U_i$ is the $i$th column of $U$ called the left singular vector of $A_w$, while $V_i$ is the $i$th column
of $V$ called the right singular vector of $A_w$.

Formula (III.14) is the essential solution of VW method, which unifies two prototype solutions
(III.2) and (III.3). Weight matrix $W$ can be arbitrarily designed as the required one. When matrix
$A_w$ is rank deficient, namely, $\sigma_n = 0$, then in general $\lambda = 0$ is not permitted in order for
the feasibility of joint velocities. However in the scheme of VW method, the lack of rank of $A_w$
often arises from the near-zero weight factors. This results in sometimes that the left singular
vector $U_i$ satisfying $U_i^Tb_w = 0$ or $U_i^Tb_w \approx 0$ associates with the zero or near-zero singular value
$\sigma_i$. In such a case, the damping factor $\lambda$ can be selected by a very small value (even zero when
$U_i^Tb_w = 0$). In other words, when $U_i^Tb_w \approx 0$, the related term $\frac{\sigma_i U_i^Tb_w}{\sigma_i^2 + \lambda^2} V_i$ can be simply cut from
(III.14), and subsequently the corresponding singular value $\sigma_i$ should not be considered for the
design of damping factor $\lambda$. 
It should be emphasized that the salient property of \( U_i^T b_w \approx 0 \) is closely related to the VW method, for which, we have the following qualitative lemma,

**Lemma 1:** Given matrix \( A_w \), when \( W = 0 \) its left singular matrix \( U \) has the form of

\[
U = \begin{bmatrix}
\tilde{U} & 0 \\
0 & S
\end{bmatrix}
\]  \hspace{1cm} (III.15)

where \( \tilde{U} \in \mathbb{R}^{m \times m} \) is just the left singular matrix of the Jacobian matrix \( J \) and \( S \in \mathbb{R}^{r \times r} \) is an orthogonal matrix.

**Proof:** When \( W = 0 \),

\[
A_w A_w^T = \begin{bmatrix}
JJ^T & 0 \\
0 & 0
\end{bmatrix}
\]  \hspace{1cm} (III.16)

Noticing that the columns of \( U \) and \( \tilde{U} \) are the orthogonal eigenvectors of \( A_w A_w^T \) and \( J J^T \), respectively, and that both of \( A_w A_w^T \) and \( J J^T \) have the same nonzero eigenvalues when \( W = 0 \), it is easy to show (III.15).

**Remark 1:** With \( b_{f_i} = 0 \), \( U_i^T b_w \to 0 \) for \( i > m \) as \( W \to 0 \), since the left singular matrix is continuous with respect to the weight matrix \( W \). This means that when \( W \) approaches to zero, the last \( n - m \) singular values of \( A_w \) will not cause the infeasibility of the obtained joint velocities although they will also approach to zeros. This explain why the small damping factor \( \lambda = 10^{-8} \) work well regardless of the near zero singular value \( \sigma_3 = 10^{-11} \) at the case of \( w_1 = 10^{-10} \) and \( w_2 = 10^{-10} \), as stated in Example 1.

Define the ratio variable \( r_i = \frac{U_i^T b_w}{\sigma_i} \). Without the damping factor, namely, \( \lambda = 0 \), the uniform solution of VW method (III.14) can be rewritten by

\[
\dot{q}_w = \sum_{i=1}^{n} r_i V_i.
\]  \hspace{1cm} (III.17)

It is more precise to say that the ratio variable \( r_i \), instead of the eigenvalue \( \sigma_i \), determines whether an impractical high value of control command happens. In other words, damping factor \( \lambda \) is not required when \( r_i \) is very small even if \( \sigma_i \) is small. In this sense, we can define the effective singular value set of \( A_w \) as

\[
\Omega_1 = \{\sigma_i : r_i > \epsilon_1\},
\]  \hspace{1cm} (III.18)

and the associated minimal effective singular value (MESV) is

\[
\sigma_{me} = \min_{\sigma \in \Omega_1} \sigma.
\]  \hspace{1cm} (III.19)

Now we are ready to present the algorithm of the VW method.
s1) calculate constraint task function $f_i(q)$ and its velocity $\dot{f}_i(q)$ for all $i$;

s2) calculate weight factor $w_i = g(f_i, \dot{f}_i, \epsilon_f)$ to build weight matrix $W$;

s3) calculate Jacobian matrices $J$ and $J_{fi}$ of the main task and constraint tasks to build matrix $A_w$;

s4) do the SVD of $A_w$ to obtain matrices $U$, $V$ and singular values $\sigma_i$, $i = 1, 2, \cdots, n$;

s5) with $\epsilon_1$, obtain set $\Omega_1$ and MESV $\sigma_{me}$;

s6) determine damping factor $\lambda$ according to the following formula [27]

$$
\lambda = \begin{cases} 
0 & \text{if } \sigma_{me} \geq \epsilon_2 \\
\left(1 - \left(\frac{\sigma_{me}}{\epsilon_2}\right)^2\right)\lambda_{max}^2 & \text{or else}
\end{cases}
$$  

s7) calculate the joint velocities by

$$
\dot{q}_w = \sum_{i: \sigma_i \in \Omega_1} \frac{\sigma_i U_i^T b_w}{\sigma_i^2 + \lambda^2} V_i.
$$

Remark 2: Because ratio $r_i$ sometimes will change sharply such that the value of $\sigma_{me}$ will experience a jump in some cases, especially when weight factor $w_i$ changes sharply. This will cause a jerk of velocity commands $\dot{q}_w$. To overcome this drawback, the following low pass filter is used before the step (s7) to smooth the damping factor,

$$
\lambda(k) = 0.9\lambda(k - 1) + 0.1\lambda
$$

where $\lambda(k)$ and $\lambda(k - 1)$ denote the damping factors of the current control loop and the last control loop, respectively, and $\lambda$ is from (III.20). According to (III.1), it can be seen that this filter does not influence the performance of active subtasks whose weights factor is far larger than $\lambda(k)$, but influences the tracking errors of main task and the feasibility of control commands.

When MESV $\sigma_{me}$ increases to larger than threshold $\epsilon_2$, the low pass filter will prolong the phase of damping, and subsequently the performance of tasks will be deteriorated but the control commands is feasible. When $\sigma_{me}$ reduces to less than threshold $\epsilon_2$, the low pass filter will retard the damping of control commands. But this retarding will not cause the infeasibility of control commands when threshold $\epsilon_1$ is designed as a very small value. The reason is at the time that $\sigma_{me}$ enters the region that $\sigma_{me} < \epsilon_2$, $r_i$ associated with $\sigma_{me}$ is close to $\epsilon_1$ by the definition of $\Omega_1$, and then by (III.17) the control commands will be feasible.

According to the definition of effective singular value set $\Omega_1$, a small threshold $\epsilon_1$ causes that some small eigenvalues that are not necessary for the design of damped factor are considered to influence the performance of tasks. Particularly, when $\epsilon_1 = 0$, the control law becomes that of the DLP method. While a larger threshold $\epsilon_1$ might cause an inadequate damping so that the
control commands are infeasible. So there is a tradeoff in the design of $\epsilon_1$, although we prefer a smaller one.

C. Damping factor and weight factors

There are three scalars, $\epsilon_1$, $\epsilon_2$ and $\lambda_{\text{max}}$ involved in the algorithm of the above subsection, as well as the function $g(f_i, \dot{f}_i, \epsilon_{fi})$ to determine the weight factors. Their selections are addressed in this subsection.

Formula (III.20) was proposed in Ref. [27] and then used in many literature [26], [28]. However the selections of $\epsilon_2$ and $\lambda_{\text{max}}$ are experiential and lack of theoretical analysis. Here we present a systematic analysis for them. Since the damping factor aims to reducing the norm of $\dot{q}_{\text{wr}}$, the reasonable goal is to make

$$K_1 = \frac{\sigma_i}{\sigma_i^2 + \lambda^2},$$

be as small as possible. Substitution of (III.20) into (III.23) yields

$$K_1 \leq \frac{1}{\epsilon_2} \quad \text{if} \quad \sigma_i \geq \epsilon_2$$

$$(\text{III.24a})$$

$$K_1 = \frac{\sigma_i}{\sigma_i^2 \left(1 - \left(\frac{\lambda_{\text{max}}}{\epsilon_2}\right)^2\right) + \lambda_{\text{max}}^2} \leq \frac{1}{2\sqrt{1 - \frac{\lambda_{\text{max}}^2}{\epsilon_2^2}} \lambda_{\text{max}}} \quad \text{if} \quad \sigma_i < \epsilon_2$$

$$(\text{III.24b})$$

Let $2\sqrt{1 - \frac{\lambda_{\text{max}}^2}{\epsilon_2^2}} \lambda_{\text{max}} = \epsilon_2$, then one has

$$\lambda_{\text{max}} = \frac{\epsilon_2}{\sqrt{2}}.$$

$$(\text{III.25})$$

Remark 3: Formula (III.25) gives the design rule of the parameters in (III.20). It ensures that gain $K_1$ achieves its maximum value $1/\epsilon_2$ at the entrance of the damping region of $\sigma_i \leq \epsilon_2$. In fact, with (III.25) the equality of the second term of (III.24b) holds only when $\sigma_i = \epsilon_2$ which but will never happen due to the condition of (III.24b).

The weight factor function of $g(f_i, \dot{f}_i, \epsilon_{fi})$ is designed as

$$w_i = g(f_i, \dot{f}_i, \epsilon_{fi}) = \begin{cases} 
\left(\frac{2\epsilon_{fi}}{f_i} - 1\right)^d & \text{if } f_i \leq 2\epsilon_{fi} \text{ and } \dot{f}_i \leq 0 \\
0 & \text{or else}
\end{cases}$$

$$(\text{III.26})$$
where \( d \) is a positive scalar to adjust the increasing speed of \( g \) with respect to \( f_i \).

When constraint function \( f_i \) is far away from the dangerous region, \( w_i = 0 \) and the constraint task is inactive; when \( f_i \) enters into the two times of the dangerous region with a negative velocity, its weight factor \( w_i \) will be not zero and the constraint task is actived, and subsequently, the velocity of \( \dot{f}_i \) begins to be repressed; when \( f_i \) further reduces to the edge of the dangerous region, namely, \( f_i = \epsilon f_{r} \), the active constraint task will have the same weight as that of the main task. At this point, \( \dot{f}_i \) will approximately have the half value of what it should be under the main task. When \( f_i \) reduces again so as to be near zero, weight factor \( w_i \) will quickly increase to infinity such that \( \dot{f}_i \approx 0 \) and the constraint function stops. This stop action will hold until \( \dot{f}_i > 0 \), that is, the constraint function starts to go away from 0.

**Remark 4:** As shown above, \( g() \) is not a continuous function. It even probably experiences a big jump when \( \dot{f}_i \) changes its sign from negative to positive. Notice that the value of \( \dot{f}_i \) is jointly determined by the main task and its constrain task according to their weights, that is, \( \dot{f}_i = \frac{\dot{f}_i^m}{1 + w_i} \) where \( \dot{f}_i^m \) is the value under the case of \( w_i = 0 \). Clearly, \( \dot{f}_i \) changes sign only when \( \dot{f}_i^m \) changes sign. Due to the continuity of \( \dot{f}_i^m \), \( \dot{f}_i^m \approx 0 \) and subsequently \( w_i^2 \| J_{\dot{f}_i}(\dot{q}) \dot{\dot{q}} \| \approx 0 \), when \( \dot{f}_i \) changes sign. Thus the discontinuous weight factor function (III.26) does not affect the continuity of the joint velocities.

**Remark 5:** Mostly in the DLP method, only the minimal singular value of \( J \) is used for the design of damped factor. The minimal singular value might be estimated by numerical filtering and without the SVD computation [29]. However in presented VW method, a complete SVD including the singular vectors and the singular values is required. It is known in general that the SVD is computationally expensive and can not be used for the real-time control. This drawback has been weaken as the speed of computer increases quickly. For certain manipulators where the Jacobian matrix is able to be decomposed into several submatrices of the low order, some analytical expressions for the SVD has been derived such that the SVD is applicable for the real time control [30], [31]. In the following experiments of 7-DOF manipulators, the numerical computation of SVD is adopted and costs 300\( \mu \)s, which is permissible for the real-time requirement of playing Ping-Pong.

### D. Performance analysis

This subsection will analyze the performance of the proposed VW method. Notice that joint velocity \( \dot{q}_w \) in (III.4) equals to \( \dot{q}_{v} \) in (II.4). It is clear that when all constraint tasks are inactive, the VW method tries best to execute the main task, as same as that of LMP. When there are
some constraint tasks active, the performance analysis seems complicate. Here for simplicity, we assume that only one constraint task \( J_{f_1} \dot{q} = 0 \) is active. In such a case, the optimal joint velocities should solve the following problem,

\[
\min \| J \dot{q} - \dot{x}_d \|, \text{ subject to: } J_{f_1} \dot{q} = 0. \tag{III.27}
\]

Let \( J_{f_1}^\perp \in \mathbb{R}^{(n-1) \times n} \) be the orthogonal complement matrix of \( J_{f_1} \), then the solution of the problem in (III.27) is

\[
\dot{q}_m = J_{f_1}^\perp (J_{f_1}^\perp J J_{f_1}^\perp)^{-1} J_{f_1}^\perp \dot{x}_d. \tag{III.28}
\]

In order for the well-posing of the above formula, the damping factor is similarly used to obtain

\[
\dot{q}_m = J_{f_1}^\perp (J_{f_1}^\perp J J_{f_1}^\perp + \lambda^2 I_{n-1})^{-1} J_{f_1}^\perp \dot{x}_d. \tag{III.29}
\]

Lemma 2: Given a manipulator with the main task of \( J \dot{q} = \dot{x}_d \) and one constraint task of \( J_{f_1} \dot{q} = 0 \), the joint velocity \( \dot{q}_w \) of the VW method satisfies

\[
\dot{q}_w \to \dot{q}_m \text{ as } w_1 \to \infty. \tag{III.30}
\]

Proof: Noting that \( \bar{W}_b = [\dot{x}_d^T, 0]^T \), expanding of (III.2) yields

\[
\dot{q}_w = (J^T J + w_1^2 J_{f_1}^T J J_{f_1}^\perp + \lambda^2 I_n)^{-1} \dot{x}_d. \tag{III.31}
\]

Define the orthogonal matrix \( B \in \mathbb{R}^{n \times n} \) and the matrix \( C \in \mathbb{R}^{n \times n} \), respectively,

\[
B = \begin{bmatrix} J_{f_1}^\perp \\ J_{f_1} / \|J_{f_1}\| \end{bmatrix}, \tag{III.32}
\]

and

\[
C = B (J^T J + w_1^2 J_{f_1}^T J J_{f_1}^\perp + \lambda^2 I_n) B^T = \begin{bmatrix} \chi_1 & J_{f_1}^T J_{f_1}^\perp \\ J_{f_1}^\perp J_{f_1}^T / \|J_{f_1}\| & \chi_2 \end{bmatrix},
\]

where \( \chi_1 = J_{f_1}^\perp J J_{f_1}^\perp + \lambda^2 I_{n-1} \) and \( \chi_2 = J_{f_1}^\perp J J_{f_1}^\perp / \|J_{f_1}\| + w_1^2 \|J_{f_1}\|^2 + \lambda^2 \). By the formula of inverse matrix of partition matrix [32], it follows that

\[
C^{-1} \to \begin{bmatrix} \chi_1^{-1} & 0 \\ 0 & \chi_2^{-1} \end{bmatrix}, \quad \text{as } w_1 \to \infty, \tag{III.33}
\]

by which one has
\[ \dot{q}_w = B^T C^{-1} B J^T \dot{x}_d = \left( J_{f_1}^{\perp T} \chi_{1}^{-1} J_{f_1}^{\perp} + \chi_{2}^{-1} \right) J^T \dot{x}_d \rightarrow \dot{q}_m. \] (III.34)

The proof is thus completed.

Remark 6: Lemma 2 shows that the VW method have the desired performance as that claimed in the end of Section II with the difference proportional to \( w_i^{-2} \). If \( m + r = n \), then \( A_w \) is a square matrix. If the weighted factors are such that \( A_w \) is far away from ill conditionedness, that is, \( \lambda = 0 \), then formula (III.2) produces \( \dot{q}_w = (A_w)^{-1}b \), by which both the main task and the constraint task are achieved without errors.

Before to ending this section, it is worthwhile to point out that the proposed VW method is a uniform scheme for the multiple tasks and is able to contain any tasks that can be represented in the level of velocity. Recalling the optimal index given in (III.1), one can think that the last term in (III.1) arises from the constraint task \( \dot{q} = 0 \) with the weight factor \( \lambda \). In fact, the VW method is also qualifiable for the control in the joint space where the Jacobian matrix of main task becomes an unit matrix. As an application, one possible way to solve the repeatability problem that only happens on the redundant manipulator is adding an extra task \( \dot{q}_i = -k(q_i - q_{0i}) \) with the associated weight \( w_{qi} \) being a sufficiently large when the manipulator returns to the start point \( q_0 \) and being zero or else, where \( q_{0i} \) is the start angle of the \( i \)th joint and \( k \) is a positive scalar selected duely.

The number of the contained tasks of VW method in theory is infinite. If there are \( n \) independent constraint tasks \( (b_{f_i} = 0) \) to be active simultaneously with \( w_i \) being sufficiently large, the VW method will give out \( \dot{q}_w = 0 \) and subsequently stop the manipulators, because there are not a direction along which the manipulator motion is permitted.

**IV. Experiment**

We carried out the experiment of the VW method on the Ping-Pong manipulator of version II of our lab, as shown in Fig. 1. The manipulator has seven degree of freedoms and its Denavit-Hartenberg (D-H) parameters, together with the permissible regions of joints, are shown in Table I.

All joints are driven by the DC brushless or DC brush motors from Maxonmotor Company, model RE-30, REmax-21, and Eci-40. They are connected with the harmonic drives, e.g., model CSF-14-100, CSF-5-100 and CSD-17-100, to drive the joints. The server amplifier ACK-055-06, made by Copley Company, working in the velocity mode drives the motor. Joint angles are
Fig. 1. Ping-Pong manipulator of version II

<table>
<thead>
<tr>
<th>Joints</th>
<th>DH-α</th>
<th>DH-a(mm)</th>
<th>DH-d(mm)</th>
<th>Joint regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>0</td>
<td>232</td>
<td>[−158, 90]</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>[75, 278]</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>0</td>
<td>280</td>
<td>[−26, 162]</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>−13</td>
<td>0</td>
<td>[68, 287]</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>[71, 258]</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>42</td>
<td>0</td>
<td>[86, 269]</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
<td>0</td>
<td>75.7</td>
<td>[−60, 300]</td>
</tr>
</tbody>
</table>

measured by the MR Encoder series from Maxonmotor Company, which is linked to the motor shaft. The workstation is a computer with an Inter(R) Core(TM)2 Quad CPU Q8200 2.33 GHz and 4GDDR2. Two controller area network (CAN) PCI cards (model PCI9820, Zlg Company) are installed to provide four CAN channel by which the computer connects the amplifiers. The Microsoft VC++ procedure executed on the Winxp operating systems provides the velocity command $\dot{q}_w$, which is sent by the CAN bus to the server amplifiers to drive the motors. The multiple threads technique is adopted, together with the Quad CPU kernels and four CAN channels, to reduce the control-loop time, whose average in our test is 2.7 ms \(^1\).

\(^1\)This is same as that for GWLN method in Ref. [19]. Although the computation cost for the transformation matrix of virtual joints of GWLN method is no longer required here but the SVD computation spends 300µs computation cost.
The initial configuration of manipulator is

\[ q_0 = [-85 \ 90 \ 85 \ 260 \ 110 \ 180 \ 150], \]  

(IV.1)

and the corresponding pose of end-effector is

\[ P_0 = \begin{bmatrix}
0.1230 & -0.0710 & 0.9899 & 0.3952 \\
-0.9696 & -0.2212 & 0.1046 & -0.1418 \\
0.2115 & -0.9727 & -0.0960 & -0.3431 \\
0 & 0 & 0 & 1.0000
\end{bmatrix}. \]  

(IV.2)

The target pose of end-effector is

\[ P_f = \begin{bmatrix}
1 & 0 & 0 & 0.3500 \\
0 & 0 & 1 & 0.1500 \\
0 & -1 & 0 & -0.0500 \\
0 & 0 & 0 & 1.0000
\end{bmatrix}. \]  

(IV.3)

The main task is to drive the end effector from the initial pose \( P_0 \) to the target pose \( P_f \) at the instant of time \( t = 2s \). The desired trajectory of the end-effector is planned with the fifth-order polynomial function, from which the desired position \( x_d \) and velocity \( \dot{x}_d \) can be given at any instant.

### A. Joint limits

Fig. 2. (a) Velocity commands (the thick solid lines) and the real velocity commands (the thin solid lines); (b) the tracking errors \( x - x_d \) of the main task; (c) the trajectory of \( y_4 \) defined in Subsection IV-B.

In the first example, only the joint limits is considered. Clearly, there are seven joint limits as the necessary constraints to be considered in the motion. In order to manifest the ability of the presented VW method on multiple constraints, here each joint limits are represented by two
constraint functions as shown in (II.8). In summary, the 14 constraint functions have the form of

\[
\begin{align*}
    f_i(q) &= q_i - q_{imin}, \\
    f_{(7+i)}(q) &= q_{imax} - q_i,
\end{align*}
\]

where \(q_i\), \(q_{imin}\) and \(q_{imax}\) are the joint angle, low bound and upper bound of the \(i\)th joint, respectively. The corresponding Jacobian matrices and velocity of constraint functions are

\[
\begin{align*}
    J_{f_i}(q) &= e_i, \\
    J_{f_{(7+i)}}(q) &= -e_i
\end{align*}
\]

where \(e_i \in \mathbb{R}^7\) is the \(i\)th row vector of the unit matrix.

For the selection of threshold \(\epsilon_{f_i}\), it should be small enough so that the chance to activate the subtask avoiding joint limits is as small as possible; while on the other hand, it should be large enough so that the joint motor have enough distances to stop its motion before the joint limits under a growing weights, that is, in theory the following inequality

\[
\epsilon_{f_i} > \frac{q_{i\epsilon}^2}{2a_{imax}},
\]

should be arbitrarily satisfied, where \(q_{i\epsilon}\) is the joint velocity at the time when the joint enters the dangerous region and \(a_{imax}\) is the maximum acceleration that the motor can provide for the joint. However, it will cause chattering to select a threshold \(\epsilon_{f_i}\) dependent on the joint velocity. Here for simplicity, we select a large enough threshold for all joints, \(\epsilon_{f_i} = 5\), for all \(i\). Let the power index \(d = 2\), then the weighted factors for the joint limits can be build from formula (III.26),

\[
w_i = \begin{cases} 
(10 \frac{f_i}{f_i} - 1)^2 & \text{if } f_i \leq 10 \text{ and } \dot{f}_i \leq 0 \\
0 & \text{or else}
\end{cases}
\]

(IV.7)
The ratio threshold to determine the set of the effective eigenvalues of $A_w$ is set as $\epsilon_1 = 0.01$. The eigenvalue threshold for damping factor is set as $\epsilon_2 = 0.1$, and the parameter $\lambda_{\text{max}}$ is determined by (III.25). In order to reduce the tracking error, a closed-loop PD control law is provided for the main task,

$$\dot{x} = \dot{x}_d - k(x - x_d), \quad \text{(IV.8)}$$

where the feedback gain $k = 5$.

The experiment results are shown in Fig. 2 and Fig. 3. In the process, only the fourth joint tends to its positive limitation, and subsequently activates the 11th constraint function $f_{11}(q)$ with a nonzero weight $w_{11}$. The weight $w_{11}$ increases as the $f_{11}(q)$ further decreases to prevent $q_4$ from touching its positive limitation. Once $f_{11}(q)$ turns back, namely $\dot{f}_{11} > 0$, the weight $w_{11}$ will immediately drop to zero.

From the Fig. 3(c), it can be seen that the nonzero damping factor can be divided into two kinds. In the first kind, which exists most times, the damping factor arises from the second smallest eigenvalue $\sigma_6$ because the smallest eigenvalue $\sigma_7 = 0$ associates with $U_7^T b = 0$ and subsequently is excluded by the effective eigenvalue set $\Omega_1$, which is defined by (III.18). The second kind happens in the time interval where the nonzero $w_{11}$ emerges such that the smallest eigenvalue $\sigma_7$ is contained in the set $\Omega_1$. In this case, the damping factor arises from $\sigma_7$ and there is a sharp change of damping factor since $\sigma_{\text{me}}$ switches from $\sigma_6$ to $\sigma_7$ that is very close to zero. After the 11th constraint becomes inactive again, all the weights are zeros and the smallest eigenvalue $\sigma_7$ is excluded from the set $\Omega_1$ again. Since $\sigma_6$ is close to the eigenvalue threshold 0.1 so that the damping factor is small. This ensures the good performance of the main tasks, as supported by the Fig. 2. The tracking error of the end point is

$$\|x - x_d\| = 1.7 \times 10^{-3}. \quad \text{(IV.9)}$$

**B. Both joint limits and obstacle avoidance**

In this experiment, there is an obstacle parallel to the $x - z$ plane expressed by $y = -0.38$ at the table surface, which means that the $y$ coordinate of all the points of manipulator should not less than $-0.38$ in the process. According to the last experiment, it is found that the origin of the 4th coordinates attached on the 4th joint in the D-H formulation can be thought as the representation point with the minimal $y$ element. Denote by $y_4$ the $y$ element of the origin of the 4th coordinates. Here beside the 14 joint limitation constraints formulated as same as that of last subsection, the obstacle constraint can be formulated by

$$f_{15} = y_4 + 0.32. \quad \text{(IV.10)}$$
Fig. 4. (a) Velocity commands (the thick solid lines) and the real velocity commands (the thin solid lines); (b) the tracking errors $x - x_d$ of the main task; (c) the trajectories of $y_4$; the blue solid line is that in the experiment of Subsection IV-B, the red dashed line is that in the experiment of Subsection IV-A.

Fig. 5. (a) the second smallest singular value $\sigma_6$ of $A_w$; (b) the smallest singular value $\sigma_7$ of $A_w$; (c) the damping factor $\lambda$; (d) the blue solid line denotes the weight factor $w_{11}$ of the constraint function $f_{11}(q)$ and the green dashed line denotes the weight factor $w_{15}$ of $f_{15}(q)$ for the obstacle avoidance.

Set the threshold and power index by $\epsilon_{f_{15}} = 0.03$ and $d = 2$, respectively. Then the weight factor $w_{15}$ has the form of

$$ w_{15} = \begin{cases} \left( \frac{0.06}{f_{15}} - 1 \right)^2 & \text{if } f_{15} \leq 0.06 \text{ and } \dot{f}_{15} \leq 0 \\ 0 & \text{or else} \end{cases} \quad \text{(IV.11)} $$

With the same PD control law as that in (IV.8), the experimental results are shown in Fig. 4 and Fig. 5. It can be seen that there is a time interval in which both $w_{11}$ and $w_{15}$ are simultaneously nonzero. For the obstacle avoidance, Fig. 4(c) and Fig. 5(d) show that once $f_{15}$ enters the dangerous region and continuously decreases, the weight factor $w_{15}$ increases accordingly to make $y_4$ to decrease slowly such that $y_4$ will not reach $-0.32$ to avoid the obstacle collision. As a comparison, $y_4$ will exceed $-0.32$ in the last experiment of Subsection IV.A without the
C. With subtasks

As stated above, the presented VW method is a uniform scheme for the multiple tasks and is able to contain any tasks that can be represented in the level of velocity. In the second experiment in last subsection where both the joint limits and obstacle avoidances are active, the tracking error is slightly bigger than that in the first experiment. Observe that the $f_{15}$ decreases as the 2th joint moves in the positive direction, we in this experiment adds a subtask to drive the 2th joint in the negative direction in such a way that $q_2$ moves from 90, the initial position, to 86. The desired trajectory of $q_2$ for this subtask is planned by a cubit spline function, by which the desired position $q_s$ and velocity $\dot{q}_s$ can be given at any instant. Thus, the subtask can be formulated by

$$J_{16} \ddot{q} = \dot{q}_s - k_q (q_2 - q_s),$$

(IV.13)
where $J_{16} = e_2$ and $k_q = 5$. The weight factor $w_{16}$ is designed by the following time-dependent function

$$w_{16} = \begin{cases} \frac{t_1}{t_1 + t_1} & \text{if } \frac{t_1}{t_1 + t_1} > 0.001 \\ 0 & \text{if } \frac{t_1}{t_1 + t_1} \leq 0.001 \end{cases}$$  \quad (IV.14)

where $t_1 = t_f/4 = 0.5$. This function means that the subtask is active in the first quarter period, after which the subtask will quickly become inactive, as shown by the trajectory of $w_{16}$ in Fig. 7(c). The experimental results are shown in Fig. 6 and Fig. 7 with the same PD control law as that in (IV.9).

In the first 0.6s, nonzero $w_{16}$ actives the subtask of the 2th joint, which, together with the main task, causes the 2th joint firstly move in the negative direction. This cause that $y_4$ firstly increases, as shown in Fig. 6(c). Subsequently, $y_4$ does not enter into the dangerous region and $w_{15} = 0$ holds in the process. Of importance is that with this extra subtask only the main task is working after approximately the instant $t = 1.2s$, although the constraint function for the negative joint limit of the 5th joint has been ever activated, which were inactive in aforementioned two experiments. In the last half phase, $\sigma_6$ is the smallest effective singular value and the damping factor $\lambda$ is very small. This causes the tracking error of the end point is smaller than that in last experiment,

$$\|x - x_d\| = 2.1 \times 10^{-3}.$$  \quad (IV.15)

V. CONCLUSION

A new VW method for the control of redundant manipulators with multiple constraints that can be formulated as subtasks in the level of velocity, is proposed in this paper. Each constraint
subtask has a varied weight factor. The zero weight factor happens when the constraint is inactive; while nonzero weight factor will prevent the constraint from being violated when the constraint is active. The VW method is a uniform scheme for the kinematic control of manipulators and is able to contain any tasks that can be represented in the level of velocity.

The weight factor rules and the damping factor rules are presented for the implementation of VW method. The weight factor is designed in such a way that at every time only the main task and the active subtasks are considered. For the damping factor, a new concept of the effective singular value is proposed to cope with the pseudo-singularity due to the transition of weight factors so as to guarantee the performance of main task as well as possible. The experiments on the 7-DOFs Ping-Pong manipulators verify the efficacy of the proposed VW method.

The proposed method has been applied for playing Ping-Pong, the main goal of the project of the Human Ping-Pong robot granted by the National 863 Program of China. At the current stage, the manipulator can continuously play with human who are able to play back the ball to a predefined region. The corresponding medias are available online. The authors ongoing work tries to extend this region by adding a movable base to manipulator.

REFERENCES


http://eelab.zju.edu.cn/xiang/robotmedia.html


Ji Xiang (M’09) received his PhD degree in control science and engineering from the Zhejiang University, hangzhou, China, in 2005. From 2005 to 2007, he worked as a postdoctoral researcher in Zhejiang University and visited City University of Hong Kong for three months during that time. Under the support of Gleddens Senior Visiting Fellowship, he has visited the University of Western Australia in 2008. He is currently an associated professor with the Department of System Science and Engineering, Zhejiang University. His research interests include complex networks, multi-agent systems and robotic manipulators.

Congwei Zhong received the B.S. degree in electrical engineering and Ph.D. degree in control science and control engineering from Zhejiang University, Hangzhou, China, in 2005 and 2011, respectively. He is currently an engineer with China Ship Development and Design Center, Shanghai. His current research interests include robotic manipulators and power system.

Wei Wei received the B.S. and M.S. degrees in Control Theory and Control Engineering from Zhejiang University, China, in 1981 and 1984, respectively. He is currently a professor in the Department of Systems Science and Engineering, Zhejiang University. His research interests include robotics, pattern recognition and networked control system.