All kinds of singularity avoidance in redundant manipulators for autonomous manipulation

Jinhyun Kim* Giacomo Marani** Wan Kyun Chung* Junku Yuh**

* Robotics & Bio-Mechatronics Lab., Dept. of Mechanical Engineering
Pohang University of Science & Technology(POSTECH), Pohang, KOREA
Tel : +82-54-279-2844 ; Fax : +82-54-279-5899; E-mail : {pluto,wkchung}@postech.ac.kr
** Autonomous Systems Laboratory,
University of Hawaii, Honolulu, Hawaii, USA
Tel : +1-808-956-7569 ; Fax : +1-808-956-2373; E-mail : {gmarani,yuh}@eng.hawaii.edu

Abstract : There are three kinds of singularity in controlling redundant manipulators. Kinematic, algorithmic and representation singularities are those. If manipulators fall into any singularity without proper action to avoid it, the control system must go away from our desire, and we can meet a dangerous situation. Hence, we have to deal the singularities very carefully. In this paper, we describe an on-line solution for avoiding the occurrence of both algorithmic and kinematic singularities in task-priority based kinematic controllers of robotic manipulators. Representation singularity can be easily avoided by using proper representation algorithm, so, in this paper, we only consider kinematic and algorithmic singularities. The proposed approach uses a desired task reconstruction and a successive task projection in order to maintain the measure for singularity over a user defined minimum value. It shows a gain in performance and a better task error especially when working in proximity of singular configurations. It is particularly suitable for autonomous systems where an off-line trajectory control scheme is often not applicable. The advantage and performance of the proposed controller is verified by simulation works. And, the experiment with real manipulator is remaining for the future works.

Keywords : Automation; redundant system; singularity avoidance; manipulability; task priority.

1 Introduction

In the repeating tasks or simple tasks, the desired task trajectory can be planned avoiding the singular points. And, it can be also overcome by mechanical design. In the manageable area, even though the manipulator are fallen into the singular points, the damage and emergency situation can be restored manually after stopping the task. However, in the uncontrollable area–space, underwater, and etc.–, we can not desire the manual labor and the tele-operation is also limited. So, the stability over singularity is the most important in those cases. Many researchers investigated to solve kinematic and algorithmic singular problem[1, 2, 3].

In autonomous robotic systems, the subtask decomposition between position and orientation is advantageous, because it will enlarge the reachable workspace of the first-priority manipulation variable (usually position) by allowing incompleteness for the second priority subtask. The concept of task priority was introduced by Nakamura[2], into the inverse kinematics of manipulators. In his approach, the occurrence of algorithmic singularities arises from conflicts between the two subtasks, when the correspondent non-prioritized task is not feasible. In this way, the performance and error for the secondary task depend on the method used for solving the inverse kinematics of the second manipulation variable. This is usually done by using classical methods like the singularity-robust inverse(Nakamura[2]). The main disadvantages for the above approaches are a loss of performance and an increased tracking error[3] especially near singular regions. The choice of damping constant must balance the required performance and the error allowed. To eliminate the occurrence of algorithmic singularities, Chiaverini[3] proposed to solve the secondary task separately and then project it onto the null space of the first manipulation variable. This algorithm has no algorithmic singularities, however, it has always an error for the secondary task except the product of the secondary task Jacobian and the primary task Jacobian pseudo-inverse is zero.

Considering the measure of manipulability as a index function(*or measure*), the approach is suitable for avoiding kinematic singularities. And, the product of the secondary task Jacobian and the primary task null space used for algorithmic singularity measure. Based on a real-time evaluation of those measure, this method does not require a preliminary knowledge of any singular configurations. Using this algorithm, we easily estimate the performance of the given measure. The result shows a good performance near the singular configurations, as shown by simulation results.

2 Task priority based method

Nakamura[2] introduced the inverse kinematics taking into account of the priority of the subtasks. Let the manipulation variable $\boldsymbol{r}_1 \in \Re^{m_1}$ be our first priority task:

$$\boldsymbol{r}_1 = f_1(\boldsymbol{q}), \tag{1}$$

where $\boldsymbol{q} \in \Re^n$ is the robot configuration vector and $\boldsymbol{r}_1$ can be, for example, the position of the end-effector. The differential relationship of (1) is:

$$\dot{\boldsymbol{r}}_1 = \boldsymbol{J}_1(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2}$$

where $\boldsymbol{J}_1(\boldsymbol{q}) \in \Re^{m_1 \times n}$ is the Jacobian matrix of the first manipulation variable, $\boldsymbol{r}_1$. Likewise, if we have additional degrees of freedom, let the manipulation variable $\boldsymbol{r}_2 \in \Re^{m_2}$ be our secondary task:

$$\boldsymbol{r}_2 = f_2(\boldsymbol{q}), \tag{3}$$

$$\dot{\boldsymbol{r}}_2 = \boldsymbol{J}_2(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where $\boldsymbol{J}_2(\boldsymbol{q}) \in \Re^{m_2 \times n}$ is the Jacobian matrix of the secondary task, $\boldsymbol{r}_2$. Equation (2) has an infinite variety of solutions for $\dot{\boldsymbol{q}}$, whose general solution is obtained using the pseudoinverse solution of the Jacobian matrix:

$$\dot{\boldsymbol{q}} = {\boldsymbol{J}_1}^+(\boldsymbol{q})\dot{\boldsymbol{r}}_1 + \left[\boldsymbol{I}_n - {\boldsymbol{J}_1}^+(\boldsymbol{q})\boldsymbol{J}_1(\boldsymbol{q})\right]\boldsymbol{y}, \tag{5}$$

where ${\boldsymbol{J}_1}^+(\boldsymbol{q}) \in \Re^{n \times m_1}$ is the pseudoinverse of $\boldsymbol{J}_1(\boldsymbol{q})$, $\boldsymbol{y} \in \Re^n$ is an arbitrary vector and $\boldsymbol{I}_n \in \Re^{n \times n}$ indicates an identity matrix. Substituting Eq. (5) into Eq. (4), we obtain:

$$\boldsymbol{J}_2\left(\boldsymbol{I}_n - {\boldsymbol{J}_1}^+\boldsymbol{J}_1\right)\boldsymbol{y} = \dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2{\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1. \tag{6}$$

If the exact solution of $\boldsymbol{y}$ exists, Eq. (6) implies that the second manipulation variable can be realized. Generally, the exact solution doesn't exist, however, we can obtain $\boldsymbol{y}$ that minimizes $\left\|\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2{\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1\right\|$ in the least square sense by using again the pseudoinverse:

$$\boldsymbol{y} = \hat{\boldsymbol{J}}_2^+\left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2{\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1\right) + \left(\boldsymbol{I}_n - \hat{\boldsymbol{J}}_2^+\hat{\boldsymbol{J}}_2\right)\boldsymbol{z}, \tag{7}$$

$$\hat{\boldsymbol{J}}_2 = \boldsymbol{J}_2\left(\boldsymbol{I}_n - {\boldsymbol{J}_1}^+\boldsymbol{J}_1\right). \tag{8}$$

Finally, substituting Eq. (7) into Eq. (5), we obtain:

$$\begin{aligned}\dot{\boldsymbol{q}} = {}& {\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1 + \hat{\boldsymbol{J}}_2^+\left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2{\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1\right)\\ &+ \left(\boldsymbol{I}_n - {\boldsymbol{J}_1}^+\boldsymbol{J}_1\right)\left(\boldsymbol{I}_n - \hat{\boldsymbol{J}}_2^+\hat{\boldsymbol{J}}_2\right)z.\end{aligned} \tag{9}$$

If we still have remaining redundancy, let now introduce a third manipulation variable $\boldsymbol{r}_3 \in \Re^{m_3}$:

$$\boldsymbol{r}_3 = f_3(\boldsymbol{q}), \tag{10}$$

$$\dot{\boldsymbol{r}}_3 = \boldsymbol{J}_3(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{11}$$

Using again the above procedure, we obtain:

$$\dot{\boldsymbol{q}} = {\boldsymbol{J}_1}^+\dot{\boldsymbol{r}}_1 + \hat{\boldsymbol{J}}_2^+\left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2\dot{\boldsymbol{q}}_1\right) + \hat{\boldsymbol{J}}_3^+\left[\dot{\boldsymbol{r}}_3 - \boldsymbol{J}_3(\dot{\boldsymbol{q}}_1 + \dot{\boldsymbol{q}}_2)\right], \tag{12}$$

where:

$$\hat{\boldsymbol{J}}_2 = \boldsymbol{J}_2\left(\boldsymbol{I}_n - {\boldsymbol{J}_1}^+\boldsymbol{J}_1\right), \tag{13}$$

$$\hat{\boldsymbol{J}}_3 = \boldsymbol{J}_3\left(\boldsymbol{I}_n - {\boldsymbol{J}_1}^+\boldsymbol{J}_1 - \hat{\boldsymbol{J}}_2^+\hat{\boldsymbol{J}}_2\right). \tag{14}$$

Equation (12) suggests the recursive idea:

$$\left\{\begin{array}{l}\dot{\boldsymbol{q}}_i = \dot{\boldsymbol{q}}_{i-1} + \hat{\boldsymbol{J}}_i^+\left(\dot{\boldsymbol{r}}_i - \boldsymbol{J}_i\dot{\boldsymbol{q}}_{i-1}\right)\\ \hat{\boldsymbol{J}}_i = \boldsymbol{J}_i{\boldsymbol{J}_i}^n\\ {\boldsymbol{J}_i}^n = {\boldsymbol{J}_{i-1}}^n - \hat{\boldsymbol{J}}_{i-1}^+\hat{\boldsymbol{J}}_{i-1}\end{array}\right., \quad \left\{\begin{array}{l}\dot{\boldsymbol{q}}_0 = \boldsymbol{0}\\ \boldsymbol{J}_0 = \boldsymbol{0}\\ {\boldsymbol{J}_0}^n = \boldsymbol{I}_n\end{array}\right.. \tag{15}$$

In Eq. (15), if any $\hat{\boldsymbol{J}}_i$ lose rank, it means singularity condition either kinematic or algorithmic.

3 Singularity avoidance for RMRC

In [4], the authors proposed kinematic singularity avoidance algorithm using task reconstruction method. In this section the algorithm is introduces briefly.

For a given manipulation variable, a singularity-free motion may be usually achieved with an off-line path planning. However, this approach requires an a-priori knowledge of all the singular configurations of the manipulator.

The proposed method, based on a real-time evaluation of the measure of manipulability, allows moving along a singularity-free path even if the singular configurations are not preliminarily known.

The basic idea is to circumscribe singularities by moving, when approaching to them, on a hyper-surfaces where the measure of manipulability is constants. Figure 1 shows this concept in a generic bi-dimensional joint space, $\Re^2$. Let's consider only one manipulation variable like:

$$\delta\boldsymbol{q} = \boldsymbol{J}^+(\boldsymbol{q})\delta\boldsymbol{r}. \tag{16}$$

The small variation of the measure of the first task singularity, $\delta m_1(\boldsymbol{q})$, is given by:

$$\delta m_1(\boldsymbol{q}) = \frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}}\delta\boldsymbol{q} = \frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}}\boldsymbol{J}^+\delta\boldsymbol{r}. \tag{17}$$

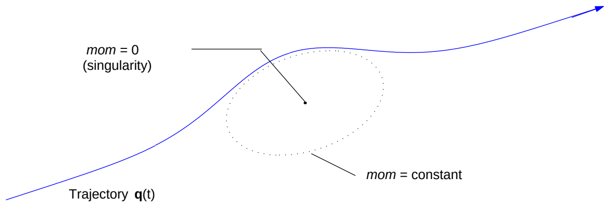


Figure 1: Singularity-free path in a generic two-dimensional joint space

In order to have $\delta m_1(\boldsymbol{q}) = 0$, Eq. (17) implies that the given task must be orthogonal to the vector

$$\frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}} \boldsymbol{J}^+ \tag{18}$$

or, equivalently, that $\delta \boldsymbol{r}$ must be lie on the surface defined by:

$$\left\{ \boldsymbol{x} \in \Re^m \ : \ \left(\frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}} \boldsymbol{J}^+ \right) \cdot \boldsymbol{x} = 0 \right\} \tag{19}$$

Let $\boldsymbol{n}_m$ be the unitary vector orthogonal to the surface Eq. (19):

$$\boldsymbol{n}_m = \frac{\left(\frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}} \boldsymbol{J}^+ \right)^T}{\left\| \frac{\partial m_1(\boldsymbol{q})}{\partial \boldsymbol{q}} \boldsymbol{J}^+ \right\|} \tag{20}$$

Then, the reconstructed task equation, $\boldsymbol{r}_p$, is given as below:

$$\delta \boldsymbol{r}_p = \delta \boldsymbol{r} - \delta \boldsymbol{r}_{corr} \tag{21}$$

where:

$$\begin{aligned} \delta \boldsymbol{r}_{corr} = & \frac{1 - sign(\delta \boldsymbol{r} \cdot \boldsymbol{n}_m)}{2} (\delta \boldsymbol{r} \cdot \boldsymbol{n}_m) \boldsymbol{n}_m k(m_1, \overline{m}) \\ & + k\left(m_1, \overline{m}/2\right) \boldsymbol{n}_m \end{aligned} . \tag{22}$$

In Eq. (22), $k(\cdot)$ represents user defined shape function. Thus, Eq. (16) becomes:

$$\delta \boldsymbol{q} = \boldsymbol{J}^+(\boldsymbol{q})(\delta \boldsymbol{r} - \delta \boldsymbol{r}_{corr}). \tag{23}$$

This algorithm can be extended as multiple task case. In other words, it can deal not only kinematic singulary but also algorithmic one.

Let's consider two subtasks case.

$$\dot{\boldsymbol{q}} = {\boldsymbol{J}_1}^+ \dot{\boldsymbol{r}}_1 + \hat{\boldsymbol{J}}_2^+ \left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2 {\boldsymbol{J}_1}^+ \dot{\boldsymbol{r}}_1 \right). \tag{24}$$

Above equation can be rewritten as

$$\dot{\boldsymbol{q}} = {\boldsymbol{J}_1}^+ \dot{\boldsymbol{r}}_1 + \hat{\boldsymbol{J}}_2^+ \hat{\dot{\boldsymbol{r}}}_2, \tag{25}$$

where the second task kinematic singularity and algorithmic singularity between the first task and the second task introduced by $\hat{\boldsymbol{J}}_2$. Therefore, for the second task singularities, $\hat{\boldsymbol{J}}_2$ does not lose rank.

Now, let's define the measure of the second task as $\delta m_2(\boldsymbol{q})$. Then, the small variation of the measure of the second task singularities, $\delta m_2(\boldsymbol{q})$, is given by:

$$\delta m_2(\boldsymbol{q}) = \frac{\partial m_2(\boldsymbol{q})}{\partial \boldsymbol{q}} \delta \boldsymbol{q} = \frac{\partial m_2(\boldsymbol{q})}{\partial \boldsymbol{q}} \hat{\boldsymbol{J}}_2^+ \delta \hat{\boldsymbol{r}}_2. \tag{26}$$

In order to have $\delta m_2(\boldsymbol{q}) = 0$, Eq. (26) implies that the given task must be orthogonal to the vector

$$\frac{\partial m_2(\boldsymbol{q})}{\partial \boldsymbol{q}} \hat{\boldsymbol{J}}_2^+. \tag{27}$$

Finally, we can get reconstructed second task as

$$\delta \boldsymbol{r}_{2p} = \delta \boldsymbol{r}_2 - \delta \boldsymbol{r}_{2corr} \tag{28}$$

where $\delta \boldsymbol{r}_{2corr}$ is given same as Eq. (22).

4 Numerical Example

In this example, we illustrate the efficiency of the proposed singularities avoidance algorithm by presenting some simulation results. The simulated manipulator is a three-link planar redundant manipulator which is the kinematic model of POSTECH DDARM–II. The desired task and workspace region are depicted in Fig. 2. Simulations for tasks of position prior to orientation are implemented. The primary task tracks $0.3m$ radius circular trajectory and the secondary task keeps $90°$ orientation of a manipulator constantly.

Some part of the primary task is lying outside of the workspace: this brings the manipulator to encounter kinematic singularities. And, some part of secondary task meets algorithmic singularities.

To avoid kinematic singularities, we use the measure of manipulability[5] as a index function:

$$m_1(\boldsymbol{q}) = \sqrt{\det \left[\boldsymbol{J} \boldsymbol{J}^T \right]} \tag{29}$$

The m_1 takes a continuous non-negative scalar value and becomes zero only when the Jacobian matrix is not of full rank. As well known, m_1 can be regarded as a distance from singularity.

It is necessary to find the derivative of Eq. (29) with respect to the joint configuration vector, $\boldsymbol{q}$. It is simply calculated as followed [6]:

$$\frac{\partial m_1(\boldsymbol{q})}{\partial q_k} = m_1(\boldsymbol{q}) \cdot \text{trace} \left\{ \frac{\partial \boldsymbol{J}}{\partial q_k} \boldsymbol{J}^+ \right\}. \tag{30}$$

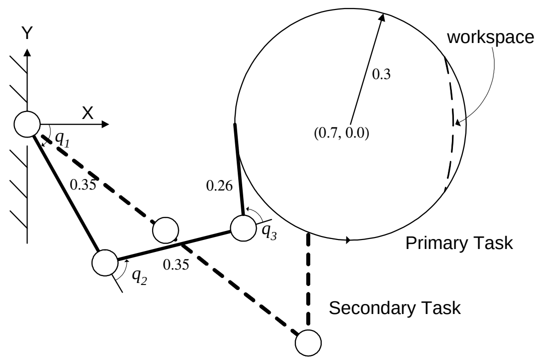


Figure 2: Desired task and workspace

Firstly, we executed 1 task case and compare our results with those of the Chiaverini's damped least-squares inverse with numerical filtering algorithm[3] which is widely used for avoiding kinematic singularity(Eq. (31)).

$$\boldsymbol{J}_{DLS}^{+} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T + \lambda^2 \boldsymbol{u}_m \boldsymbol{u}_m^T)^{-1}, \tag{31}$$

$$\lambda^2 = \begin{cases} 0 & \sigma_m \geq \epsilon \\ \left(1 - (\frac{\sigma_m}{\epsilon})^2\right) \lambda_{max}^2 & \sigma_m < \epsilon, \end{cases} \tag{32}$$

where σ_m is the lowest singular value of $\boldsymbol{J}$ and $\boldsymbol{u}_m$ is the corresponding output singular vector.

For the simulation of the proposed algorithm, the control frequency is 1kHz and $\overline{m}$ is set to 0.06. And, for the Chiaverini algorithm, ϵ and λ_{max} are set to 0.05 and 0.07, respectively. Those are tuned up for the best task performance.

Figures 3 and 4 show the result of the proposed algorithm and the Chiaverini algorithm, respectively. In Fig 3(c), the manipulability measure is maintained exactly $0.03(\overline{m}/2)$ which corresponds to the lower limit of k. So, the proposed concept of geometrical reconstruction is well performed. And, as shown in Figs. 3(b) and 4(b), the task recovery action is better using the proposed one around $t = 2.2$ for errors. The maximum x-direction errors are almost same, because the workspace boundary is 0.96m. But, the task recovery actions after escaping from the kinematic singularity region are different. Contrary to Fig. 4(b), in Fig. 3(b), the task recovered immediately without over(under)shoot.

In Fig. 4(d), the joint velocities are about two times large comparing with the proposed one, which results from the small value of manipulability (Fig. 4(c)).

From these simulation study, the performance of the task reconstruction algorithm is verified. With the manipulability measure, it guarantees the robustness near singular points.

Nextly, we ran the robot with two subtasks. In this case, the desired trajectory has both kinematic and algorithmic singularity situation. Figure 5 shows simulation results. In this figure, the proposed algorithm is robustly performed over both kinematic and algorithmic singularities.

5 Conclusions

The proposed method for kinematic and algorithmic singularity avoidance allows moving along a singularity-free path for a generic manipulator whose singular configurations are not preliminarily known. This is done without using a global approach and so it is suitable for a real-time implementation. In addition, one of the advantages of the proposed algorithm is that the performances are predictable. It allows to perform "as much as possible" the desired task under the constraint that the distance from a singular configuration must be greater than a lower limit.

Acknowledgement

This research was supported by the International Cooperation Research Program("Development of Intelligent Underwater Vehicle/Manipulator and Its Control Architecture", M6-0302-00-0009-03-A01-00-004-00) of the Ministry of Science & Technology, Korea.

References

[1] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formualtions and damped least-squares methods," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-16, pp. 93–101, Jan. 1986.

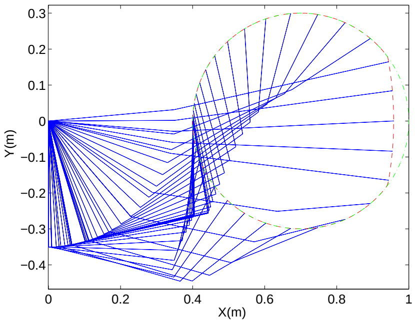
[2] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization.* Addison Wesley, 1991.

[3] S. Chiaverini, "Singularity-robust task-priority redundancy resoultion for real-time kinematic control of robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 398–410, June 1997.

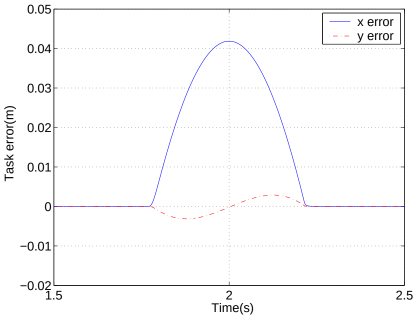
[4] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators," in *Proc., IEEE Int. Conf. on Robotics and Automation*, pp. 1973–1978, 2002.

[5] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. of Robotic Research*, vol. 4, no. 2, pp. 3–9, 1985.

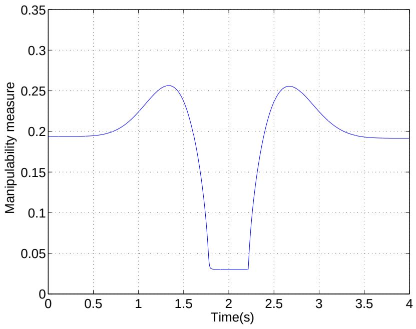
[6] J. Park, *Analysis and Control of Kinematically Redundant Manipulators: An Approach Based on Kinematically Decoupled Joint Space Decomposition.* PhD thesis, Pohang University of Science and Technology(POSTECH), 1999.



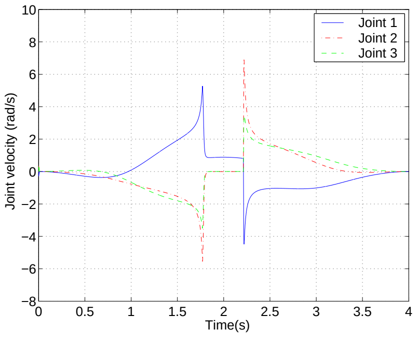
(a) Task in the x-z plane



(b) Task errors

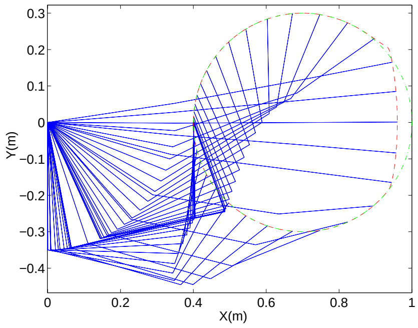


(c) Manipulability

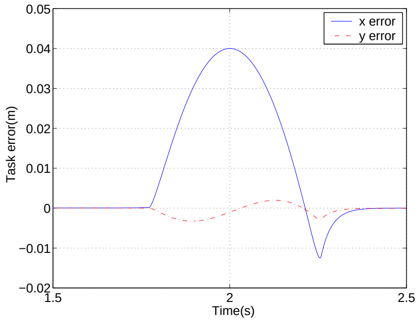


(d) Joint velocity

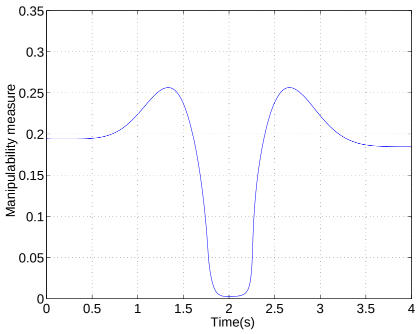
Figure 3: Simulation with the proposed algorithm



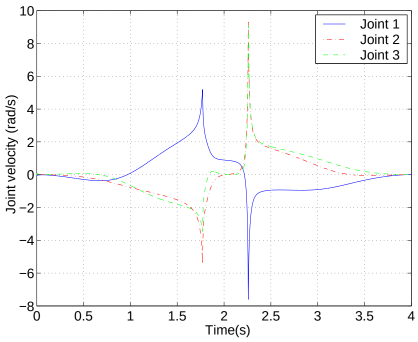
(a) Task in the x-z plane



(b) Task errors

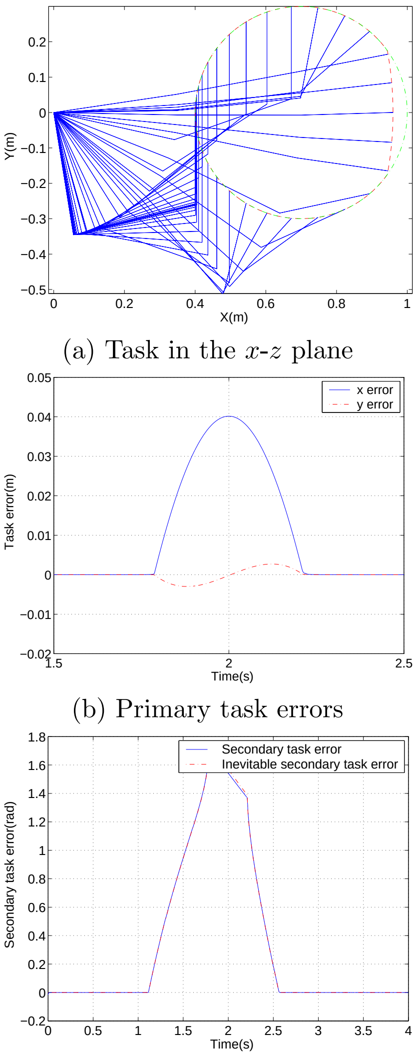


(c) Manipulability



(d) Joint velocity

Figure 4: Simulation with damped least-squares inverse



(a) Task in the x-z plane

(b) Primary task errors

(c) Secondary task error

Figure 5: Simulation result with two subtasks