Chapter from the book
Downloaded from: http://www.intechopen.com/books/

Interested in publishing with IntechOpen?
Contact us at book.department@intechopen.com
Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm

Hyejin Han and Jaeheung Park

Abstract When robots are controlled in the task space, singularities and joint limits are among the most critical and difficult issues that can arise. In this paper, we propose a new approach for the robots to operate in the regions near singularities and joint limits using the operational space control framework. Specifically, a continuous task transition algorithm called the intermediate desired value approach is applied to the hierarchically structured controller in the operational space control framework. In this approach, new tasks are defined for dealing with singularities and joint limits, and the tasks are activated or deactivated using the continuous task transition algorithm to guarantee the continuous execution of the tasks during the execution of the main task. The proposed approach is implemented on a 6-DOF manipulator called Roman-MD. The experimental results demonstrate its performance near the singular regions and joint limits.

Keywords Joint Limit Avoidance, Singularity, Operational Space Control, Task Transition

1. Introduction

The workspace of robots is limited by singularities and joint limits. When robots are controlled by task space control algorithms, it is necessary to handle singularities and joint limits during the execution of tasks so as to perform the tasks successfully (Figure 1) [1].

There have been various approaches to operating robots near singularities and joint limits in task space control. To deal with singularities in the kinematic control, a damped least-square inverse Jacobian matrix was proposed to...
resolve the singularity problem [2–4]. It provided an approximate solution when the robot was close to a singular position. Later, the transpose of the Jacobian was used instead of the inverse, which also approximates the solution [5, 6]. In addition, a redundancy-control method based on a task-priority strategy was proposed as a means of kinematic control [7–9]. This approach allows robots to move near or even pass through singular configurations. Moreover, a method is proposed to control the robot to stop and switch its controller when it is detected that the robot is near a singular region in a kinematic control scheme [10, 11]. On the other hand, there have been few studies on this issue in the framework of operational space control. Some of the solutions were specifically designed for the singularity types on PUMA650 robots [12, 13].

Meanwhile, to overcome the problem of joint limits, a method was proposed based on a weighted least norm solution for a redundant robot in the kinematic control [14, 15]. The weighted least norm solution was also compared with the gradient projection approach [16], showing that the weighted least norm method can avoid unnecessary self-motion and oscillation when the correct magnitude of self-motion is used. In line with redundancy resolution algorithm, quadratic programming (QP) methods have been proposed in the kinematic control [17] and the QP-based dynamical system approach has been developed [18].

For the operational space control framework, a typical solution is to use the classical potential field approach [19]. However, it is often difficult to select appropriate gains because the selection depends on the configuration and the controller of the existing tasks. Moreover, it was pointed out that the potential field methods have some problems such as local minimum and oscillations [20–22].

A solution for dealing with unilateral constraints, including both singularity and joint limit, was proposed for the kinematic control [23, 24], which modifies the null-space projection matrices for continuous control law during the transition among multiple tasks. This approach was further developed to be applied in the operational space control framework [25]. Similarly, a method using continuous null space projections was proposed for torque-controlled robots [26]. Both of the methods are based on the modification of the null space projection matrix and limited in the use of multiple hierarchical structures.

Recently, a task transition approach called the intermediate desired approach was proposed to deal with task transition among multiple tasks in kinematic control [27] and was further developed to be used in the operational space control framework [28]. This approach is different from the previously mentioned approaches [25, 29] in that the desired values are modified while the control structure remains the same. Additionally, multiple constraints and tasks can be performed with multiple levels of priorities, which is important when dealing with joint limit avoidance, singularities and tasks in different priorities. In this paper, the problems of singularity and joint limit avoidance are resolved by this intermediate desired value approach in the operational space control framework.

The main contribution of this paper compared to our previous works is that the present study provides complete details pertaining to issues relating to singularity and joint limit tasks during the implementation of the intermediate desired value approach in the operational space while also demonstrating its feasibility via its performance on a physical robot.

The main idea of using the intermediate desired value approach when dealing with singularities and joint limits is first to define the tasks corresponding to the singularities and joint limits, and then to insert or remove these tasks while the main task of the robot is executed. The intermediate desired value approach provides a smooth transition among multiple tasks with or without a hierarchy. Specifically, constraint tasks, such as joint-limit-avoidance tasks, are provided with high priority over the primary task in order to conduct tasks accurately and to maintain uninterrupted transitions with a continuous task transition algorithm.

This approach is implemented both in simulation and on a physical robot which is a 6-DOF manipulator called Roman-MD. In the experiments of joint limit avoidance, the proposed approach is compared with the potential field approach [19]. Then, the operation of the robot is demonstrated approaching near singularity and coming back to the initial position. Finally, the control of the robot is performed in the configuration near both joint limit and singularity.

The paper is organized as follows. First, the task transition algorithm in the operational space control framework is summarized in Section 2. Second, stability during transitions is discussed in Section 3. Singularity and joint limit avoidance using the task transition algorithm are presented in Section 4 and 5, respectively. The experimental results are plotted and discussed in Section 6. The paper is concluded in Section 7.

2. Continuous Task Transition Approach

We approach the problems of handling singularities and joint limits by applying a continuous transition algorithm [27]. In the following sub-sections, we first recall a hierarchical control structure in the operational space control framework and then introduce a continuous task transition approach.

2.1. Hierarchical Control Structure

When a robot is controlled in the operational space control framework, the task space dynamics is described as follows: [30]

\[ \Lambda_t \dot{x}_t + \mu_t + p_t = F_t, \]  

and

\[ \Lambda_t = (J_t A_t^{-1} J_t^T)^{-1}, \]  
\[ \mu_t = \Lambda_t (J_t A_t^{-1} b - J_t q), \]  
\[ p_t = \Lambda_t J_t A_t^{-1} g. \]

Here, \( q, A, b \) and \( g \) are the vectors of the joint angles, the joint space mass/inertia matrix, the Coriolis/centrifugal
torque, and the gravity torque, respectively. The terms $J_t$, $\Lambda_t$, $\mu_t$, $p_t$ and $F_t$ are the Jacobians corresponding to the task $x_t$, the task space inertia matrix, the vector of Coriolis/centrifugal forces, the vector of gravity forces, and the control force for this task. The control torque for control of the task can then be composed using the task dynamics [30].

$$\Gamma_i = J_t^T F_i = J_t^T \{A_t f_i^* + \mu_t + p_t\}$$

$$= J_t^T \Lambda_t \{f_i^* + \eta_i + \zeta_i\} = J_t^T \Lambda_i \tilde{f}_i,$$

and

$$\eta_i = \Lambda_i^{-1} \mu_i, \quad \zeta_i = \Lambda_i^{-1} p_i.$$  

In these equations, $f_i^*$ is the control input for the unit mass system for the task $x_t$.

At this point, when the robot has a constraint task, such as a joint limit avoidance task, we consider the constraint task as $x_c$ and the primary task as $x_t$ in a hierarchical structure. In other words, the command torque for the primary task is controlled in the null space of the constraint task as follows [30]:

$$\Gamma = \Gamma_{\text{constraint}} + \Gamma_{\text{task}}$$

$$= J_t^T F_t + N_t^T \Gamma_t$$

$$= J_t^T F_t + N_t^T (J_t^T F_t)$$

$$N_t^T = I - J_t^T J_t^T$$

$$\tilde{\Lambda}_{c,t} = A^{-1} J_t^T \Lambda_t.$$  

where $J_c$ and $J_t$ are the Jacobians corresponding to the constraint and primary tasks, respectively. The matrix $J_t$ is the dynamically consistent inverse of the Jacobian $J_c$. The terms $F_c$ and $F_t$ are the control forces corresponding to $x_c$ and $x_t$, respectively. The term $N_t^T$ is the null space projection matrix of the constraint task. The dynamics in the constraint and task space can then be derived [30]:

$$\Lambda_c \ddot{x}_c + \mu_c + p_c = F_c$$  

$$\Lambda_{c,t} \ddot{x}_t + \mu_{c,t} + p_{c,t} = F_t$$  

where

$$\Lambda_{c,t} = (J_c N_c A^{-1} N_t^T J_t^T)^{-1}$$

$$\mu_{c,t} = \Lambda_{c,t} (J_c A^{-1} b - J_c \dot{q})$$

$$p_{c,t} = \Lambda_{c,t} J_c A^{-1} g.$$  

Using the equations of motion in the constraint and task spaces, the control forces $F_c$ and $F_t$ can be formulated as follows:

$$F_c = \Lambda_c f_c^* + \mu_c + p_c = \Lambda_c \tilde{f}_c$$

$$F_t = \Lambda_{c,t} f_t^* + \mu_{c,t} + p_{c,t} = \Lambda_{c,t} \tilde{f}_t.$$  

If the two equations are combined, the control torque for the hierarchical task set is

$$\Gamma = J_t^T \Lambda_c \tilde{f}_c + N_t^T J_t^T \Lambda_{c,t} \tilde{f}_t.$$  

When a constraint with a high priority is added or removed, the torque command switches between (3) and (12). The task transition approach is introduced in the next section to resolve the issue of a discontinuity in such cases.

2.2. Task Transition Approach for both Prioritized Tasks and Non-prioritized Tasks

The continuous task transition algorithm known as the intermediate desired value approach is applied for a smooth transition when new tasks are inserted or removed. A joint limit avoidance task, for example, can be created with a high priority level while a primary existing task is executed. Therefore, we can revise equation (12) by applying the intermediate desired value approach.

In addition, for dealing with singularity, some parts of an existing task can be selected for removal, in which case the task removal or insertion transition is at the same hierarchical level as the existing task. Moreover, we need an algorithm to deal with operations that avoid both joint limits and singularities simultaneously.

For one typical example, consider the execution of one constraint task with two tasks with equal priorities: $x_c$, $x_{t1}$ and $x_{t2}$. By applying the intermediate desired value approach, the term $\tilde{f}_c$ is modified to $\tilde{f}_c^*$ as follows:

$$\Gamma = J_t^T \Lambda_c \tilde{f}_c^* + N_t^T J_t^T \Lambda_{c,t} \tilde{f}_t^*$$

where

$$J_t = \begin{bmatrix} J_{t1} \\ J_{t2} \end{bmatrix}, \quad \tilde{f}_t = \begin{bmatrix} \tilde{f}_{t1} \\ \tilde{f}_{t2} \end{bmatrix}$$

and

$$N_t^T = I - J_t^T J_t^T$$

$$\Lambda_{t,c}^{-1} = J_t N_c A^{-1} N_t^T J_t^T.$$  

The intermediate values are defined by

$$\tilde{f}_{t1} = h_{t1} f_{t1} + (1 - h_{t1}) \tilde{J}_{t1}^T \Lambda_{t,c}^{-1} \Gamma_{t1}$$

$$\tilde{f}_{t2} = h_{t2} f_{t2} + (1 - h_{t2}) \tilde{J}_{t2}^T \Lambda_{t,c}^{-1} \Gamma_{t2}.$$  

These equations provide a continuous transition among prioritized tasks when the task $x_c$ is inserted at a higher priority. The activation parameters $h_{t1}$, $h_{t2}$ and $h_{t2}$ change from 0 to 1 continuously for a smooth transition when the task is inserted, whereas they change to 0 when the task is removed. The term $\Gamma_{t1/c}$ denotes the torque input using the intermediate desired value excluding the $n$ task:

$$\Gamma_{t1/c} = J_t^T \Lambda_c \tilde{f}_{t1}$$

where

$$\tilde{f}_{t1} = h_{t1} \tilde{f}_{t1} + (1 - h_{t1}) \tilde{J}_{t1}^T \Lambda_{t,c}^{-1} \Gamma_{t1}$$

$$\tilde{f}_{t2} = h_{t2} \tilde{f}_{t2} + (1 - h_{t2}) \tilde{J}_{t2}^T \Lambda_{t,c}^{-1} \Gamma_{t2}.$$  

In the equations above, in order to compute $\Gamma_{t1/c}$, we need to consider task $x_{t1}$ and $x_{t2}$, which have the same priority level. On the other hand, in the cases of $\Gamma_{t1/1}$ and $\Gamma_{t2/1}$, we take into account task transitions with priority. Therefore, $\Gamma_{t1/1}$ can be computed as

$$\Gamma_{t1/1} = J_t^T \Lambda_c \tilde{f}_{t1} + N_t^T J_t^T \Lambda_{c,t} \tilde{f}_{t2}$$

where

$$\tilde{f}_{t1} = h_{t1} \tilde{f}_{t1} + (1 - h_{t1}) \tilde{J}_{t1}^T \Lambda_{t,c}^{-1} \Gamma_{t1}$$

$$\tilde{f}_{t2} = h_{t2} \tilde{f}_{t2} + (1 - h_{t2}) \tilde{J}_{t2}^T \Lambda_{t,c}^{-1} \Gamma_{t2}.$$  

Hyejin Han and Jaeheung Park: Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm

www.intechopen.com
3. Stability During Transition

The proposed approach uses the operational space control framework with a hierarchical structure having priorities. The control structure is not modified in this approach, so the controlled system is stable if not in transition. The modelling uncertainties may affect the performance of the system, but these are mostly overcome by the feedback on the unit-mass system in the operational space control framework.

Now, the stability during transition is discussed in this section. The intermediate desired value approach is applied to the control inputs \( \tilde{f} \) and \( \tilde{f}_i \) in (12), which correspond to constraint and task. During transition, the intermediate desired values of the control inputs, \( \tilde{f}_i \) and \( \tilde{f}_i \), are computed using (16). Each intermediate desired value of the control input in (16) is composed of two parts: the control input of the constraint or task itself and the effect from the control of other constraints or tasks. They are scaled down by \( h_i \) or \( h_t \), and \((1-h_i)\) or \((1-h_t)\), respectively. As the activation parameter decreases, the control of the constraint or task itself diminishes. That is, the DOF of the system used for the control of the corresponding constraint or task is gradually used by the other constraints or tasks.

Considering this constitution of the sub-systems, their stability depend on each controller with the control input of the constraint or task itself. In this paper, a PD control is designed for each sub-system, which behaves as a unit mass system by using the operational space control framework if not in transition [31]. The proportional and derivative gains can be chosen to have critically damped responses for each unit mass system. Then, when the task is in transition, the effect of using the activation parameters is to decrease the proportional and derivative gains with the same ratio. Therefore, the response of each system in transition becomes over-damped with a lower bandwidth. Since the system becomes over-damped, the system is still stable with the comprised bandwidth of the task.

4. Joint Limit Avoidance

As stated in the Introduction section, there has been much research on implementing stable operation near joint limits or on avoiding joint limits. The task transition approach can also be very effective in the application of a joint limit avoidance scheme in the operational space control framework.

We need to manage the robot so that it moves within the range of joint movement while the main task \( a_k \) is performed. This can be accomplished by inserting a joint limit avoidance task as a new task with a higher priority level than the other tasks when one of the joints comes close to its lower or upper joint limits. We set the joint limit buffer regions considering each joint limit, and a new task of avoiding the joint limit is created within this buffer region considering the range of joint motion [27]. In this buffer region, we treat the formation and destruction of joint limit avoidance task as cases of task insertion and removal, respectively. Consequently, we need a smooth task transition during the performance of these tasks.

Considering the task of avoiding the joint limit, the torque command is computed as

\[
\Gamma = J^T_f A_f \tilde{f}_f + N^T_f \Gamma_f
\]

where

\[
\tilde{f}_f = h_f \tilde{f} + (1 - h_f) A^{-1}J^T_f A_f \tilde{f}_f \\
\Gamma_f = J^T_f A_{jl} \tilde{f}_f.
\]

In these equations, \( J_f \), \( F_f \) and \( N_f \) are the Jacobian, the control force for joint limit avoidance, and its null-space projection matrix, respectively, while \( \Gamma_f \) is the torque for the main task. For example, if the fourth joint is in the buffer region of its joint limit, \( J_f \) is defined as follows for a 6-DOF manipulator:

\[
J_f = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}
\]

Moreover, to maintain smooth operation while the joint limit avoidance task is active or inactive, the activation parameter of the joint limit avoidance task varies continuously between 0 and 1 as a function of the corresponding joint angle. The activation parameter increases as the corresponding joint angle approaches the joint limit. In this experiment, the activation function in an earlier study [27] is used, as plotted in Figure 2. Specifically, the activation function \( h_i \) is designed as follows:

\[
h_i = \begin{cases} 
1.0, & \text{if } q_i \geq q_{ij} \\
f(q_i), & \text{if } q_i < q_i < q_{ij} \\
0.0, & \text{if } q_i \leq q_i \leq q_{ij} \\
g(q_i), & \text{if } q_i < q_i < q_{ij} \\
1.0, & \text{if } q_i \geq q_{ij}
\end{cases}
\]

and

\[
f(q_i) = 0.5 + 0.5 \sin \left( \frac{\pi}{\beta} (q_i - q_{ij}) - \frac{\pi}{2} \right) \]

\[
g(q_i) = 0.5 + 0.5 \sin \left( \frac{\pi}{\beta} (q_i - q_{ij}) + \frac{\pi}{2} \right)
\]

The \( i \)th joint angle, \( q_i \), is constrained by the lower limit \( q_{ij} \) and the upper limit \( q_{ij} \), and \( \beta \) denotes the size of the activation buffer

\[
q_i = q_{ij} + \beta.
\]
On the other hand, the activation parameter for the primal task is maintained at 1 throughout the execution of the task because it is always fully activated.

The size of the buffer region can be determined depending on the specific task. If the size is large, the transition will be more smoothly conducted. However, the normal workspace would be smaller and the transition time would be longer. The size is set to approximately 10% of the joint range, 0.15 rad, in this experiment.

Note that the dimension of the Jacobian for the joint limit depends on how many joints are in the buffer region. The activation parameters for each joint limit task, which correspond to each row in $J_{jl}$, have different values because the joints are at different points in their buffer regions. For example, if the first and fourth joints are in transition for a 6-DOF robot, we have two joint limit avoidance tasks $x_{j1}$ and $x_{j2}$ at the same priority level. The Jacobian can be formulated as follows:

$$J_{jl} = \begin{pmatrix} J_{j1} \\ J_{j2} \end{pmatrix}$$

(27)

where

$$J_{j1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

(28)

The transitions among these joint limit avoidance tasks can be regarded as non-prioritized tasks in equation (13). Above all, the joint limit avoidance task $x_{jl}$ needs to have a higher priority than the primal task $x_l$.

5. Singularity Avoidance

In this section, we discuss the implementation of singularity avoidance using the intermediate desired value approach in the operational space control framework. The idea is to deactivate the task along the singular direction near the singularity. The singular value decomposition (SVD) is used on the inverse of the task space inertia matrix to identify the singular directions and to define the task associated with it. The activation parameter for continuous transition is designed to change based on the singular value which is lower than a certain threshold.

First, singularity is constantly monitored by the singular value decomposition (SVD) of the inverse of the inertia matrix of the primary task $x_l$.

$$\Lambda^{-1}_t = J_t A^{-1} J_t^T = U_t \Sigma_t U_t^T$$

(29)

Here, $\Sigma_t$ is a diagonal matrix with singular values, and $U_t$ is a rotation matrix [30, 32].

When one of the singular values is less than a threshold, the corresponding direction is considered as a singular direction. Therefore, $U_t$ is decomposed into two parts: $U_{ns}$ for non-singular directions and $U_s$ for the singular direction.

$$U_t = \begin{bmatrix} U_{ns} & U_s \end{bmatrix}$$

(30)

The primary task $x_l$ is then decomposed into two tasks as follows:

$$x_{l,ns} = U_{ns}^T x_l$$

and

$$x_{l,s} = U_s^T x_l$$

(31)

The corresponding Jacobians are defined as

$$J_{l,ns} = U_{ns}^T J_l$$

and

$$J_{l,s} = U_s^T J_l$$

(32)

Accordingly, task $x_{l,ns}$ and task $x_{l,s}$ are at the same priority level and the control torque is computed as

$$\Gamma = J_l^T \Lambda_t f_l$$

(33)

where

$$J_l = \begin{pmatrix} J_{l,ns} \\ J_{l,s} \end{pmatrix}, \quad f_l = \begin{pmatrix} \hat{f}_{l,ns} \\ \hat{f}_{l,s} \end{pmatrix}$$

(34)

and

$$\hat{f}_{l,ns} = h_{l,ns} (1 - h_{l,ns}) J_{l,ns} A^{-1} J_{l,ns}^T \hat{f}_{l,ns}, \quad \hat{f}_{l,s} = h_{l,s} (1 - h_{l,s}) J_{l,s} A^{-1} J_{l,s}^T \hat{f}_{l,s}$$

(35)

In the following experiment of controlling the end-effector as a main task, the orientation task is given a higher priority level than the position. This is because the singular values and the corresponding directions have more intuitive physical meaning when the position and orientation tasks are separately considered [33]. Thus, it is proper to divide the tasks initially into two main parts: the position and the orientation.

Herein, a hierarchical structure, which is similar to a constraint-task structure, is applied and the task of controlling the end-effector is divided into two parts (13). The subscripts $o$ and $p$ indicate the orientation task and the position task, respectively. When the singular direction comes up from the position task, the control torque can be formulated as

$$\Gamma = J_l^T \Lambda_o f_l + N_l^T J_l^T \Lambda_{p,o} f_p$$

(36)

where

$$J_p = \begin{pmatrix} J_{p,ns} \\ J_{p,s} \end{pmatrix}, \quad f_p = \begin{pmatrix} \hat{f}_{p,ns} \\ \hat{f}_{p,s} \end{pmatrix}$$

(37)

and

$$\hat{f}_{p,ns} = h_{p,ns} (1 - h_{p,ns}) J_{p,ns} A^{-1} \Gamma_{p,ns}, \quad \hat{f}_{p,s} = h_{p,s} (1 - h_{p,s}) J_{p,s} A^{-1} \Gamma_{p,s}$$

(38)

The activation parameter for task $x_{l,ns}$ is kept at 1 because we want to execute the task fully along the non-singular direction. Thus, task $x_{l,s}$ along the singular direction is activated or deactivated by the activation parameter which is designed as a function of the corresponding singular value. The activation parameter for the singular value starts with 1 at the threshold value and then decreases as the singular value decreases such that the control of the task in that direction will be progressively deactivated. One example of such an activation parameter is plotted in Figure 3, where the threshold is set to 0.02 and decreases until the value reaches 0.01. These values and the shape of the function can be designed by considering the physical intuition of the inertial property of the manipulator.
At this point, when we consider both a joint limit and a singularity simultaneously, the joint limit avoidance task has a higher priority. Moreover, as already stated, task $x_i$ is initially decomposed into the orientation task $x_o$ and the position task $x_p$ when singularity avoidance is considered. Hence, the order of task priorities is the joint limit task, the task for orientation control $x_o$, and then the position control task $x_p$.

\[
\Gamma = \Gamma^i_{\beta} + \Gamma^o_{\gamma} + \Gamma^p_p
\]  

where \[
\Gamma^i_{\beta} = J^i_{p} \Lambda^i_{\beta} \hat{f}^i_{\beta} \\
\Gamma^o_{\gamma} = (J_{p} N_{\alpha})^T \Lambda^o_{\gamma} \hat{f}^o_{\gamma} \\
\Gamma^p_p = (J_{p} N_{\alpha})^T \Lambda^p_{\alpha} \hat{f}^p_p
\]

and

\[
J_{p} = (J_{p,ns})^T, \quad f_{p} = (J_{p,ns} f_{p,ns})^T.
\]

The intermediate values are defined as follows:

\[
f^i_{\beta} = h_{\beta} \hat{f}^i_{\beta} + (1 - h_{\beta}) J^i_{\beta} A^{-1} \Gamma^i_{\beta/[i]} \\
f^o_{\gamma} = h_{\gamma} \hat{f}^o_{\gamma} + (1 - h_{\gamma}) (J_{p} N_{\alpha})^T A^{-1} \Gamma^o_{\gamma/[p,ns]} \\
f^p_{p,ns} = h_{p,ns} \hat{f}^p_{p,ns} + (1 - h_{p,ns}) (J_{p,ns} A^{-1} \Gamma^p_{p/[p,ns]}) \\
f^p_{p,s} = h_{p,s} \hat{f}^p_{p,s} + (1 - h_{p,s}) (J_{p,s} A^{-1} \Gamma^p_{p/[p,s]}).
\]

6. Experimental Results

The proposed approach was implemented on a 6-DOF manipulator called Roman-MD (Figure 4). The robot is controlled using software called RoboticsLab [34]. The servo rate was set to 500 Hz when reading the encoders and sending the torque commands to the motors. In the following experiments, the control input for the tasks is composed by PD control.

\[
f^* = k_{p} (x^d - x) - k_{v} \dot{x},
\]

where $x^d$ and $x$ are the desired value, the current state and the derivative of the current state of the task. The terms $k_{p}$ and $k_{v}$ are the proportional and derivative feedback gains, respectively. Because each task becomes a unit mass system using the nonlinear feedback linearization by the operational space control framework, the gains can be selected based on the task specifications. In this paper, the closed-loop frequency and damping ratio were chosen as 20.0 rad/sec and 1.0, respectively, to be a critically damped system. The gains $k_{p}$ and $k_{v}$ are then 400.0 and 40.0 because it is a unit mass system.

6.1. Joint Limit Avoidance Task

In this experiment, the position of the end-effector is controlled to move to the lower-right direction and come back to the initial position while the orientation of the end-effector is controlled to be the initial orientation, as illustrated in Figure 5. The first joint limit avoidance task is inserted while the positioning task is performed.

In order to avoid the joint limit, the activation parameter of the first joint increases at around three seconds, as the first joint enters its joint limit buffer region, as shown in Figure 6.

In this experiment, the upper and lower limits for all joints are set to 0.78 rad and $-0.78\text{rad}$, respectively. These chosen values are smaller than necessary to show the performance of the proposed algorithm effectively. Furthermore, the size of the buffer region is 0.15 rad, and the activation parameter in Figure 2 is used.

The experiment demonstrates that the joint limit task is activated and deactivated without any abrupt change in the control of the end-effector (Figure 7). The position of the robot cannot be controlled as desired when the constraint task of joint limit avoidance is activated.

In the next experiment, we compared the potential field approach and the proposed task transition approach for joint limit avoidance through the simulation using the 6-DOF manipulator. The control input $f_i$ using the potential field approach [19] is,

\[
f_i = \begin{cases} 
\eta \left( \frac{1}{\rho_i} - \frac{1}{\rho_{i(0)}} \right) \frac{1}{\rho_i^3}, & \text{if } \rho_i \leq \rho_{i(0)} \\
0.0, & \text{if } \rho_i > \rho_{i(0)} \quad \text{(44)}
\end{cases}
\]

where

\[
\rho_i = q_i - \bar{q}_i \\
\rho_{i(0)} = \bar{q}_i - q_i.
\]
Figure 5. Snapshots from the joint limit avoidance experiment.

Figure 6. Joint velocity, joint angle and activation parameter of the first joint during the joint limit avoidance experiment.

Figure 7. Plots of end-effector position during the joint limit avoidance experiment.

Figure 8. Simulation: execution of the joint limit task using the potential field approach and the proposed continuous control approach.

Figure 9. Fourth joint angle and end-effector position during the joint limit avoidance simulation using the potential field approach.

Here, \( \rho_{i}(0) \) and \( \rho_{i}(0) \) are the distance limit of the potential field influence. The \( i \)th joint angle is \( q_{i} \), and \( q_{i} \) and \( q_{i} \) represent the lower limit and the upper limit.

In this simulation, the desired position is set out of the joint range. While the task is performed, the fourth joint reaches its limit buffer region (Figure 8). In the case of the potential field approach, the constant gain \( \eta \) was chosen as 0.5, and the distances \( \rho_{i}(0) \) and \( \rho_{i}(0) \) are 0.08 and -0.08, respectively.

As shown in Figure 9 and Figure 10, the potential field approach, in comparison with the proposed approach, can easily cause oscillation and instability unless the gains are carefully tuned. Although the oscillatory response of the potential field approach shown in this experiment can be improved by changing the parameters of \( \eta \) and \( \rho_{i}(0) \) at the specific configuration, it is not easy to find suitable values for all the configurations of the robot [21, 22]. The oscillatory and unstable response is shown in this experiment to point out the typical problem of using the potential field approach in the application of joint limit avoidance.
Accordingly, this step can only be carried out if the matching procedure was already performed for the first error. Joint limit and singularity avoidance experiment. Int. j. adv. robot. syst., 2013, Vol. 10, 346:20138 www.intechopen.com

6.2. Handling Singularity

The experiment of controlling the robot near singularity is presented in this section. As noted in Section 5, the robot can deal with singularities by deactivating the task along the singular direction.

In this experiment, the end-effector is commanded to move from the upper-right to the lower-left position and come back to the initial position (Figure 11). The position of the end-effector cannot reach the desired position exactly because the deactivated direction is not fully controlled.

The activation parameter of the task is designed as a function of the singular value by singular value decomposition (SVD), and the buffer of the singular values is identical to that plotted in Figure 3.

Figure 12 and Figure 13 show the data plots from this experiment. The activation parameter for the singular direction starts to decrease at $\sigma = 0.02$ (around 4 seconds) and the positioning task becomes smoothly deactivated along the singular direction. Because the singular direction is on the y-z plane, the position in the z-direction does not follow the desired position trajectory when the position task in the singular direction is in transition region.

Figure 11. Experiment of the singularity avoidance.

6.3. Joint Limit and Singularity Avoidance

In this section, related to Section 5, the case is demonstrated in which both the joint limit and singularity are handled when controlling the end-effector of the robot as a primary task. As shown in Figure 14, the position of the end-effector is commanded to move toward the desired position and come back to the initial position.

As shown in Figure 15, when the singular value enters the buffer region at approximately three seconds, the activation parameter for the singular direction decreases according to the activation function of the singular value as designed. Approximately 1 second later, the task of avoiding the fourth joint limit is added to the commanding

Figure 14. Joint limit and singularity avoidance experiment.
did not follow the desired trajectory completely near the singularity.

It should be noted that the position of the end-effector during the experiment is plotted in Figure 17. The position of the end-effector when the robot is controlled near both singularity and joint limit is shown in Figure 16. The joint angle and activation parameter of the second joint when the robot is controlled near both singularity and joint limit are shown in Figure 15.

In this paper, we propose a control strategy for dealing with singularities and joint limits using the intermediate desired value approach in the operational space control framework. The intermediate desired value approach in the operational space control framework enables an effective and stable transition when tasks are inserted and removed. Controlling the robot near singularity is implemented by deactivating the task along the singular direction. In addition, the transition algorithm is effectively applied to both the insertion and the removal of the joint limit task with a hierarchical control structure. The experimental results demonstrate that the robot is successfully controlled when it approaches its joint limits and a singular region using the proposed approach. The proposed approach in this paper demonstrated the robot operation with only motion tasks. Our future work is directed toward generalizing this control approach to deal with tasks involving both motion and contact so that it can be used in robot-environment interaction.

Acknowledgement

This work was sponsored by the Basic Science Research Program (no.2010-0005799) and by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MSIP) (no.2011-0032014).

References


