Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks

J. Sverdrup-Thygeson*, S. Moe†, K. Y. Pettersen*, and J. T. Gravdahl†
*Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, NTNU,
Norwegian University of Science and Technology, Trondheim, Norway
Email: {Jorgen.Sverdrup-Thygeson, Signe.Moe, Kristin.Y.Pettersen}@ntnu.no
†Department of Engineering Cybernetics, NTNU, Norwegian University of Science and Technology, Trondheim, Norway
Email: Tommy.Gravdahl@ntnu.no

***Abstract*—This paper proposes a novel method for kinematic singularity avoidance for robot manipulators. Using set-based singularity avoidance tasks within the singularity-robust multiple task priority framework, avoidance of kinematic singularities is guaranteed without interfering with the convergence of compatible equality tasks. Non-compatible equality tasks will still be fulfilled to the extent possible. In addition, the method can be used to reconfigure the robot into a more dexterous configuration. The proposed method is applicable to both redundant and non-redundant robots, with both fixed base and floating base. The implementation is generic and independent of the type and number of tasks. Although also applicable for non-redundant fixed base robot manipulators, this novel approach is particularly well suited for highly redundant and/or floating base manipulators, since the robot configuration can be changed to improve the dexterity of the manipulator arm. We demonstrate the method by applying it to an underwater swimming manipulator, which is an innovative and highly redundant underwater floating base manipulator. Furthermore, simulation results are presented that illustrate and validate the proposed method.**

I. Introduction

Task-oriented control of a robotic mechanism is a control approach which combines the cognitive abilities of humans with the computational power of modern day computers [1]. While a human operator can provide outstanding capabilities in terms of situational awareness and decision making, the computers are more efficient when it comes to exploring multiple options to solve a given task and evaluating the suitability of each solution. Multiple solutions can occur when the robotic system is redundant. In the redundant case, the problem admits an indefinite number of solutions. This redundancy can be exploited to solve multiple tasks simultaneously, and to analyze and select the most appropriate solution based on specific constraints and optimization criteria. Inverse kinematic control, and in particular the task priority method [2], [3], has been a valuable tool for redundancy resolution of robotic manipulators for nearly three decades. The task priority approach has been utilized for obstacle avoidance [2], for handling kinematic singularities [4]–[6], and to respect joint limits [7], as well as for a number of other performance indices.

A major challenge for robot manipulators is avoidance of kinematic singularities, as these are associated with loss of or reduced motion capability of the robot end-effector and infeasible joint velocity commands. Kinematic singularities can occur both within the workspace of the robot (internal singularities) and at the edge of the workspace (external singularities). A redundant robotic mechanism can avoid internal singularities by utilizing the excess degrees of freedom of the joints, and external singularities can be handled by relocating the base of the robot. In order to characterize the closeness to kinematic singularities, several manipulability measures have been suggested, e.g. the manipulability index [8], the condition number, and the smallest singular value [9] of the manipulator Jacobian.

In [10], the damped least-squares (DLS) technique was introduced, and it was suggested to use a configuration-varying damping factor as a function of the manipulability index. The DLS method was also investigated in [11] to provide a user-defined accuracy of the solution, and in [5] in combination with the singularity-robust task priority framework. In general, the DLS technique is well suited for non-redundant manipulators, as it ensures nice behavior in near singular configurations by damping high joint velocity commands. However, the method does not guarantee that a singularity will not be reached. In [4] the task priority method is applied specifically to avoid kinematic singularities for a 7-DOF redundant manipulator, by formulating an analytic constraint task to push the manipulator away from known singular configurations. The same idea is pursued in [6], where a low priority equality task is added to guide an underwater vehicle with a 3-DOF manipulator arm away from a known singularity associated with the second joint. This method will, however, achieve avoidance of kinematic singularities only if the primary task and the secondary constraint task are

This research was funded by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 – NTNU AMOS, the Centre of Research Based Innovation, project No. 237900 – SFI Manufacturing in Norway, and by VISTA, a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Statoil.

compatible, i.e. both tasks can be satisfied simultaneously for all robot configurations. In addition, this approach assumes a priori knowledge of the singular configurations, such that proper constraint tasks can be designed.

Singularity avoidance using nonlinear optimization techniques are presented in e.g. [12] and [13], where the goal is to minimize a cost function to realize a compromise between a primary task and avoidance of kinematic singularities and joint limits. The relative priority between the tasks are determined by constant weighting factors, which must be tuned appropriately to obtain the desired behavior. The use of dynamic weights is investigated in e.g. [7], where a task priority approach is combined with fuzzy logic. Here, manipulability is considered by triggering a fuzzy rule to reconfigure the manipulator arm into a fixed dexterous configuration whenever the smallest singular value of the manipulator Jacobian is close to zero.

Another method is considered in [14], [15], where a dynamic task priority algorithm switches between different solutions based on a measure of the manipulability. This method is quite intricate and requires correction terms to be added to all the other tasks. Also, the method only avoids kinematic singularities associated with the position of the end-effector, and does not necessarily guarantee a singularity-free path for the orientation of the end-effector.

In this paper, we propose a novel method for kinematic singularity avoidance, using set-based tasks [16] within the singularity-robust multiple task priority (SRMTP) framework [5], [17], [18]. To the authors' best knowledge, no existing method can ascertain avoidance of kinematic singularities without compromising the fulfillment of the other tasks. The method proposed in this paper guarantees avoidance of kinematic singularities and convergence of all compatible task errors to zero by introducing a singularity avoidance task as a high priority set-based task [16]. The method is straightforward to implement, regardless of the type and the number of tasks. Moreover, it can be applied to maintain a dexterous configuration, such that the manipulator arm is always ready to carry out the next assignment. The proposed method handles both internal and external kinematic singularities within the same formulation. Furthermore, there is no need for prior knowledge of the singular joint configurations, as this is handled autonomously by the algorithm. This novel method is applicable to a large class of redundant robotic manipulators, and it is particularly well suited for mobile robots and floating base manipulators where the base of the robot can be moved to improve the manipulability.

To illustrate the proposed method, we apply it to an underwater swimming manipulator (USM) [19], which is an innovative underwater robotic mechanism. The USM is essentially a crossover between an autonomous underwater vehicle (AUV) and an underwater snake robot (USR) [20], [21]. The USM is a multi-body articulated structure, but unlike conventional USRs, the USM is equipped with additional thrusters, thus enabling it to operate as a floating base robotic manipulator (see Figure 1). Since the USM is a floating base manipulator arm with a fully actuated base, the Jacobian of the end-effector always has full rank. Consequently, it is always possible to relocate the base of the USM to fulfill a positioning task for the end-effector while simultaneously avoiding singular configurations of the manipulator arm. However, it is desirable to utilize the joints as much as possible to control the end-effector, due to the faster dynamic response and higher precision of the joint control. The thrusters should primarily be used to relocate the base of the USM, in order to extend the operational workspace of the manipulator arm.

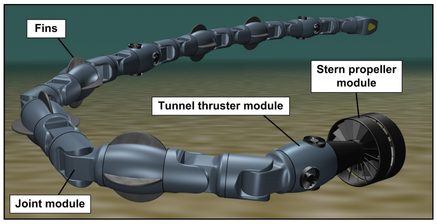


Fig. 1: Underwater swimming manipulator

In this paper, we exploit set-based tasks [16] to avoid kinematic singularities, while satisfying a position and orientation task for the head link of the USM. We investigate multiple cases in which both high-priority and low-priority set-based tasks are included in the task priority sequence.

This paper is organized as follows. The notation and the end-effector kinematics of a generic floating base manipulator is defined in Section II. Section III provides a brief review of the set-based SRMTP framework, and describes how set-based tasks can be utilized for guaranteed avoidance of kinematic singularities. In Section IV, we demonstrate the proposed method by applying it to a USM, while Section V provides simulations to illustrate the flexibility and the benefits of using both high- and low-priority set-based singularity avoidance tasks. Conclusions are given in Section VI.

II. Notation and kinematics

The state of a generic floating base robotic manipulator can be represented by the configuration variables, $\xi = \begin{bmatrix}\eta_b^T & q^T\end{bmatrix}^T \in \mathbb{R}^{6+n}$, where $\eta_b \in \mathbb{R}^6$ describes the the overall position and orientation of the base of the robot in the inertial frame, and $q \in \mathbb{R}^n$ is the n-dimensional vector of joint angles. Specifically, the orientation of the robot base is represented by the Euler angles $\eta_{b,2} = [\phi_b, \theta_b, \psi_b]^T \in \mathbb{R}^3$. The inertial frame is denoted $\mathscr{F}_0$, while the reference frames attached to the base of the robot and the end-effector are labeled as $\mathscr{F}_b$ and $\mathscr{F}_e$, respectively. The position and orientation of the end-effector is represented in the inertial frame, $\mathscr{F}_0$, by $\eta_e \in \mathbb{R}^6$, while the vector $\zeta = \begin{bmatrix}(V_{0b}^b)^T & \dot{q}^T\end{bmatrix}^T \in \mathbb{R}^{6+n}$ describes the body-fixed velocities of the robot manipulator. The relation between the body-fixed velocities and the inertial frame velocities can then be expressed by $\dot{\xi} = J_\xi(\eta_{b,2})\zeta$, where

$$J_\xi(\eta_{b,2}) \triangleq \begin{bmatrix} R_{0b}(\eta_{b,2}) & 0_{3\times 3} & 0_{3\times n} \\ 0_{3\times 3} & T_{0b}(\eta_{b,2}) & 0_{3\times n} \\ 0_{n\times 3} & 0_{n\times 3} & I_n \end{bmatrix}, \tag{1}$$

$R_{0b}(\eta_{b,2}) = R_z(\psi_b)R_y(\theta_b)R_x(\phi_b) \in \mathbb{R}^{3\times 3}$ is the rotation matrix from the inertial frame to the base frame of the robot, and

$$T_{0b}(\eta_{b,2}) = \begin{bmatrix} 1 & \sin\phi_b \tan\theta_b & \cos\phi_b \tan\theta_b \\ 0 & \cos\phi_b & -\sin\phi_b \\ 0 & \frac{\sin\phi_b}{\cos\theta_b} & \frac{\cos\phi_b}{\cos\theta_b} \end{bmatrix}. \tag{2}$$

The forward kinematics of the robot manipulator, i.e. the coordinate transformation from $\mathscr{F}_e$ to $\mathscr{F}_b$, can be found by consecutive multiplication of the coordinate transformations between the joints,

$$g_{be} = g_{b1}g_{12}\cdots g_{ne} = \begin{bmatrix} R_{be} & p_{be} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{3}$$

In (3), $p_{be}(q) \in \mathbb{R}^3$ and $R_{be}(q) \in SO(3)$ represent the relative position and orientation between the base of the robot and the end-effector, respectively. Moreover, the coordinate transformation matrices from $\mathscr{F}_b$ and $\mathscr{F}_e$ to the inertial frame, $\mathscr{F}_0$, are given by

$$g_{0b} = \begin{bmatrix} R_{0b} & p_{0b} \\ 0 & 1 \end{bmatrix}, \quad g_{0e} = g_{0b}g_{be} = \begin{bmatrix} R_{0e} & p_{0e} \\ 0 & 1 \end{bmatrix}, \tag{4}$$

respectively. The position and orientation of the base of the robot and the end-effector in the inertial frame can thus be found by $\eta_b = g_{0b}^\vee$ and $\eta_e = g_{0e}^\vee$, respectively, where the $^\vee$ notation describes the vector representation of the corresponding coordinate transformation matrix.

III. Singularity avoidance

In this section, we first give a brief presentation of how set-based tasks [16] can be included in the singularity-robust multiple task priority framework. We then propose a novel solution to the problem of kinematic singularity avoidance for a generic robot manipulator by introducing set-based singularity avoidance tasks. We show that a high priority set-based singularity avoidance task will guarantee that the measure of manipulability is kept above a specified minimum value, while a low priority set-based singularity avoidance task provides a practical and flexible way to reconfigure the robot manipulator into a more dexterous configuration. The definition of the chosen measure of manipulability is given in Section III-D, while we in Section IV show how the proposed method can be applied to a USM for guaranteed kinematic singularity avoidance.

A. The singularity-robust multiple task priority framework

When using a task priority approach, the end goal is to control the robotic mechanism such that one or more tasks are accomplished in a strictly prioritized order. Each task is described by a task variable, $\sigma(t)$, associated with either the internal configuration of the robot or the external configuration with respect to the environment. For a general robotic system, the primary task is typically to achieve a desired position and orientation of the end-effector. A comprehensive collection of possible tasks for underwater robotic vehicles is presented in [22]. In general, the relation between the configuration variables, $\xi(t)$, and the task variables, $\sigma_i(t)$, are given by the task functions, $f_i(\xi(t))$, according to

$$\sigma_i(t) = f_i(\xi(t)). \tag{5}$$

In order to determine the motion of the robot required to fulfill the different tasks, we need to solve the inverse kinematic problem associated with each task function (5). This is more conveniently done at the velocity level, i.e. by finding the first-order task Jacobians. The task Jacobians are matrices of partial derivatives that describe how a small change in the configuration of the robot will affect the task variables. Taking the time-derivative of (5) and inserting (1) yields

$$\dot\sigma_i = J_i(\xi)\zeta, \quad J_i(\xi) \triangleq \frac{\partial f(\xi)}{\partial \xi} J_\xi(\eta_{b,2}). \tag{6}$$

In this paper we use the SRMTP framework for solving the inverse kinematic problem for redundant robotic mechanisms. The generic expression for the SRMTP framework with k tasks assigned to different priority levels is

$$\begin{aligned} \zeta_{\text{ref}} = J_1^\dagger\left(\dot\sigma_{1,d} + \Lambda_1\tilde\sigma_1\right) + N_1 J_2^\dagger\left(\dot\sigma_{2,d} + \Lambda_2\tilde\sigma_2\right) \\ + \ldots + N_{12..(k-1)} J_k^\dagger\left(\dot\sigma_{k,d} + \Lambda_k\tilde\sigma_k\right), \end{aligned} \tag{7}$$

where $\sigma_{i,d}$ for $i \in [1,\ldots,k]$ represent the desired task values, and $\tilde\sigma_i = \sigma_{i,d} - \sigma_i$ are the task errors. The null space of the task Jacobians are given by $N_i = (I - J_i^\dagger J_i)$, and $N_{12..(k-1)}$ is the combined null space of tasks 1 through $k-1$, calculated by stacking the corresponding task Jacobian matrices and finding the null space of this compound matrix. The null space matrices filter out the velocity components generated by lower priority tasks that would otherwise interfere with the higher priority tasks. All the task errors will converge to zero, provided that the tasks are compatible and specified as time-independent regulation tasks, and that the task gains Λ_i are chosen properly [18]. The task priority sequence will have a large impact on the observed behavior of the robotic manipulator. Next, we briefly present how the SRMTP framework, in combination with recent results [16], provides a complete inverse kinematics framework that incorporates both equality and set-based tasks.

B. Set-based tasks in the SRMTP framework

It is common to distinguish between equality tasks, in which the objective is to make the task variables converge to specific desired values, i.e. $\lim_{t\to\infty}\sigma(t) = \sigma_d(t)$, and set-based tasks which are satisfied when the task variable is contained in its valid set, $\sigma \in \mathscr{D} = [\sigma_{\min}, \sigma_{\max}]$. Singularity avoidance, collision avoidance, constrained workspace, joint limit avoidance, and field of view are some examples of tasks for robotic manipulators that are more appropriately formulated as set-based tasks than as equality tasks.

In [16], the SRMTP framework is extended to include set-based tasks. The set-based SRMTP framework can briefly be described as follows [16]:

While σ is in the interior of the valid set, the set-based task is considered inactive and is thus ignored by the inverse kinematics algorithm. If σ reaches the boundaries of the valid set, the set-based task is activated and introduced in the task priority sequence as an additional equality task, given that it would otherwise leave the set. The set-based task is deactivated as soon as the other tasks ensure that σ stays within the valid set. Activation/deactivation of the set-based task is determined by the tangent cone function. The tangent cone, T_D, to the valid set $\mathscr{D}$ at the point $\sigma \in \mathscr{D}$ is defined as

$$T_D(\sigma) = \begin{cases} [0,\infty), & \sigma = \sigma_{\min} \\ \mathbb{R}, & \sigma \in (\sigma_{\min}, \sigma_{\max}) \\ (-\infty, 0], & \sigma = \sigma_{\max} \end{cases} . \tag{8}$$

Note that achieving $\dot{\sigma} \in T_D \ \forall t$ implies $\sigma \in \mathscr{D} \ \forall t$. Thus, we obtain the desired behavior by maintaining $\dot{\sigma} \in T_D$. The desired value for the inserted equality task is defined as the boundary value of the valid set. If the set-based task is introduced as a high-priority task, then this task will be satisfied for all time, i.e. $\sigma \in \mathscr{D} \ \forall t$, and all compatible lower priority equality tasks will converge to zero. On the other hand, if the set-based task is introduced as a low-priority task, the trajectory of the task variable, σ, may escape the valid set due to the influence of the higher priority equality tasks. However, the set-based task variable will converge towards the valid set when the set-based task is no longer in conflict with the higher priority tasks.

Each set-based task switch between active and inactive, which means that there is in general a total of 2^j different active/inactive combinations to consider, where j is the number of set-based tasks. These combinations are referred to as modes. The activation and deactivation of the set-based tasks result in an algorithm that dynamically switches between multiple solutions. The system states are stable through the switching process, and the equality task errors converge to zero given the assumptions listed before (compatible tasks, proper choice of gains, regulation tasks). Algorithm 1 outlines how the active mode is chosen in the case of a single set-based task based on the tangent cone of the valid set.

Algorithm 1: Mode selection for one set-based task.

```
while True do
    a = in_TangentCone($\sigma$, $\dot{\sigma}$, $\sigma_{\min}$, $\sigma_{\max}$);
    if a is True then
        mode = mode_1;
    else
        mode = mode_2;
    end
end
```

In the remainder of this paper, we follow the notation established in [16], and label the task variables for the equality tasks as σ_i, with $i \in [1, \ldots, k]$, where k is the number of equality tasks, and the set-based tasks with subscripts in alphabetic order, $\sigma_a, \sigma_b, \ldots$.

C. Set-based tasks for singularity avoidance

In this section we propose a novel approach to singularity avoidance for robot manipulators. In particular, we propose using the concept of set-based tasks to avoid kinematic singularities, by including a measure of manipulability as a set-based task. The goal is to keep the measure of manipulability of the robot above a minimum value, and thus avoid kinematic singularities. As explained in Section III-B, this can be guaranteed by introducing the singularity avoidance task as a high-priority set-based task.

The task variable for the singularity avoidance task is defined as $\sigma_a(q) \in [\sigma_{a,\min}, \infty)$, where $\sigma_{a,\min}$ represents the minimum allowed distance to a kinematic singularity as defined by the measure of manipulability. Note that $\sigma_a(q)$ is a function of the joint angles only, as the measure of manipulability is not concerned with the position and orientation of the robot in the inertial frame, η_b. Therefore, the proposed method is equally applicable to both fixed base and floating base robots. We find the Jacobian of the singularity avoidance task by applying (6) with $\sigma_a(q)$ as the task function

$$J_a(q) = \frac{\partial \sigma_a(q)}{\partial \xi} J_\xi(\eta_{b,2}). \tag{9}$$

The set-based singularity avoidance task is activated if the task variable σ_a is about to leave the valid set. For high-priority set-based tasks, the generic expression in (7) is then modified to

$$\begin{aligned} \zeta_{\text{ref}} = N_a J_1^\dagger \left(\dot{\sigma}_{1,d} + \Lambda_1 \tilde{\sigma}_1\right) + N_a 1 J_2^\dagger \left(\dot{\sigma}_{2,d} + \Lambda_2 \tilde{\sigma}_2\right) \\ + \ldots + N_{a12..(k-1)} J_k^\dagger \left(\dot{\sigma}_{k,d} + \Lambda_k \tilde{\sigma}_k\right), \end{aligned} \tag{10}$$

where the projection through the null space of the singularity avoidance task Jacobian, $N_a = (I - J_a^\dagger J_a)$, ensures that σ_a is frozen at the lower boundary of the valid set and will not be further reduced. Thus, the robot manipulator maintains the specified minimum distance to the kinematic singularities.

In addition, this method can be utilized to improve the dexterity of the manipulator configuration by including the singularity avoidance task as a low-priority set-based task. In this case, a strict minimum distance to kinematic singularities is not guaranteed, but the measure of manipulability will still converge towards the desired minimum value when the singularity avoidance task is active and when it is compatible with the higher priority equality tasks. We refer to [16] for further details on the theoretical background and stability analysis of set-based tasks in general.

In the next section we will discuss the selection of the measure of manipulability, while we illustrate this novel approach to singularity avoidance by applying it to the particular case of an underwater swimming manipulator in Section IV.

D. Measures of manipulability

Various measures of manipulability have been proposed in previous literature, in order to design singularity robust methods and to avoid kinematic singularities altogether. In this section, we go through some of them, and in particular, we explain our choice of manipulability measure to include in the set-based SRMTP approach to ensure that the robot avoids kinematic singularities.

Let s_i, for $i \in [1,\dots,m]$, denote the singular values of $J_M J_M^T$, where $J_M(q) = \frac{\partial \eta_{be}(q)}{\partial q}$ is the Jacobian of $\eta_{be}(q) = g_{be}^{\vee} \in \mathbb{R}^6$, which represents the relative position and orientation between the base and the end-effector. One measure of manipulability is the manipulability index, which is given by [8]

$$w = \sqrt{\det(J_M J_M^T)} = s_1 s_2 \dots s_m. \tag{11}$$

The manipulability index is proportional to the volume of the m-dimensional manipulability ellipsoid. As the manipulator arm approaches a singular configuration, one of the singular values tends to zero and the volume of the manipulability ellipsoid also converges towards zero. This indicates loss of the ability to change the pose of the end-effector in one degree of freedom. As such, it is desirable to keep the manipulability index above a specified minimum value.

Another measure of manipulability is the condition number,

$$\kappa = \frac{s_{\max}}{s_{\min}}, \tag{12}$$

which is the ratio of the largest to the smallest singular value of $J_M J_M^T$. The condition number describes the shape of the manipulability ellipsoid, and when the condition number $\kappa = 1$, the manipulability ellipsoid is isotropic. The use of the condition number can be useful, as a measure of manipulability, if the goal is to attain an isotropic joint configuration, i.e. a configuration in which the ability to move the end-effector is equal in all degrees of freedom.

The smallest singular value, $s_{\min}$, can also be used as a measure of the distance to a singular configuration. This measure of manipulability is typically used in configuration-varying damped least squares methods [9].

In this paper, we select the manipulability index squared, $w^2 = \det(J_M J_M^T)$, as the measure of closeness to singular configurations, because this expression can be calculated analytically and also allow us to find an analytical expression for the task Jacobian for the singularity avoidance task. With this selection, we specify the singularity avoidance task as

$$\sigma_a = w^2 \in \mathscr{D} = [\sigma_{a,\min}, \infty). \tag{13}$$

We can use the square of the manipulability index since we are only interested in positive values. Moreover, the square function is a convex function, which means that when the proposed method keeps the square of the manipulability index above a specific value, this will also be the case for the actual manipulability index.

In the next section, the task functions and the associated task Jacobians are derived for the specific case of an underwater swimming manipulator.

IV. Case study - Underwater swimming manipulator

In this section, we demonstrate how the proposed method for kinematic singularity avoidance can be applied to a floating base robotic manipulator. More specifically, we implement the set-based method for a USM, which is a self-propelled underwater manipulator arm. With a fully actuated base and a high number of joints, the USM is highly redundant, and thus well suited for application of the set-based kinematic singularity avoidance method. The base of the USM is specified to be located at the back of the tail link, and the tip of the head link is considered as the end-effector. In this case study, we consider the motion of the USM in a horizontal plane, because the set-based singularity avoidance concept is satisfactorily illustrated in 2D. The USM considered in this paper consists of five links connected by four 1-DOF joints rotating around the local z axis. The joint angle vector is thus given by $q = [q_1, q_2, q_3, q_4]^T$.

In addition to the set-based singularity avoidance task, we consider the following two equality tasks for the USM,

- achieving a desired position and orientation of the end-effector, and
- achieving a desired position of the USM base.

For robot manipulators in general, achieving a desired end-effector pose is an essential task. As the USM is a floating base robot manipulator, it is also desirable to include the desired position of the base as an equality task, as this can reduce the overall motion of the USM and increase the utilization of the joints.

A. End-effector task

The task variables and the desired values for the end-effector tasks are represented by

$$\sigma_1 = \begin{bmatrix} x_e & y_e \end{bmatrix}^T, \qquad \sigma_{1,d} = \begin{bmatrix} x_{e,d} & y_{e,d} \end{bmatrix}^T, \tag{14}$$

$$\sigma_2 = \psi_e, \qquad \sigma_{2,d} = \psi_{e,d}. \tag{15}$$

The corresponding task errors are $\tilde{\sigma}_1 = \sigma_{1,d} - \sigma_1$ and $\tilde{\sigma}_2 = \sigma_{2,d} - \sigma_2$. For the planar case, the transformations between the joints of the USM are given by

$$g_{i(i+1)} = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & l_i\cos(q_i) \\ \sin(q_i) & \cos(q_i) & 0 & l_i\sin(q_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

where $i \in [1,\dots,n]$ and l_i is the length of each link. As outlined in Section II, the task variables can be expressed by

$$\sigma_1 = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} g_{0e}^{\vee}, \tag{17}$$

$$\sigma_2 = \psi_e = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} g_{0e}^{\vee}, \tag{18}$$

and the corresponding task Jacobians are given by (6)

$$J_1 = \frac{\partial \sigma_1(\xi)}{\partial \xi}\begin{bmatrix} R_z(\psi_b) & 0_{3\times 4} \\ 0_{4\times 3} & I_{4\times 4}\end{bmatrix}, \tag{19}$$

$$J_2 = [0,0,1,1,1,1,1]. \tag{20}$$

B. Base position task

The task variable and the task Jacobian for the base position task are given by

$$\sigma_3 = \begin{bmatrix} x_b & y_b \end{bmatrix}^T \tag{21}$$

$$J_3 = \begin{bmatrix} \cos(\psi_b) & -\sin(\psi_b) & 0_{1\times 5} \\ \sin(\psi_b) & \cos(\psi_b) & 0_{1\times 5}\end{bmatrix} \tag{22}$$

C. Singularity avoidance task

The singularity avoidance task variable is expressed by

$$\begin{aligned}\sigma_a =&\det(J_M J_M^T) \\ =&l_2^2 l_3^2 + l_2^2 l_4^2 + l_3^2 l_4^2 - l_2^2 l_3^2 \cos(2q_2) - l_3^2 l_4^2 \cos(2q_3)\ldots \\ &- l_2^2 l_4^2 \cos(2q_2+2q_3) - l_2 l_3 l_4^2 \cos(q_2+2q_3)\ldots \\ &- l_2^2 l_3 l_4 \cos(2q_2+q_3) + l_2 l_3 l_4^2 \cos(q_2) + l_2^2 l_3 l_4 \cos(q_3),\end{aligned} \tag{23}$$

where J_M is the manipulator Jacobian defined in Section III-D. The task Jacobian, J_a, is given by (9). Equation (23) shows that $\sigma_a = 0$ and the USM is in a singular configuration when $q_2 = q_3 = 0$.

V. Simulations

In this section, we perform simulations to illustrate the application of the proposed set-based approach for singularity avoidance for a USM. As discussed earlier, defining the singularity avoidance task as a high priority set-based task guarantees by [16] that this task will be achieved and that the task errors of compatible equality tasks will converge to zero. No such guarantee can be made when using a low priority set-based task. However, using a low priority set-based singularity avoidance task ensures that the USM will reshape itself into a more dexterous configuration. To illustrate how the behavior depends on the chosen order of priority between the tasks, both high-priority and low-priority set-based tasks are investigated, one by one and combined. The end-effector position and heading tasks are specified as regulation tasks with $\dot{\sigma}_{1,d} = \dot{\sigma}_{2,d} = 0$ and changing setpoints, while the desired base position is equal to the initial position, $\sigma_{3,d} = \begin{bmatrix} x_{b0} & y_{b0}\end{bmatrix}^T$. The minimum value for the singularity avoidance task variable is set to $\sigma_{b,\min} = 0.8$ for the low-priority case and to $\sigma_{a,\min} = 0.4$ for the high-priority case. In these simulations, the end-effector position and heading tasks are assigned the same priority, i.e. they are combined into a single equality task. In order to get a realistic visual interpretation of the results, we utilize the simulation tool Vortex by CM Labs [23] for visualization of the kinematic motion of the USM. Videos of the four different cases are available at [24].

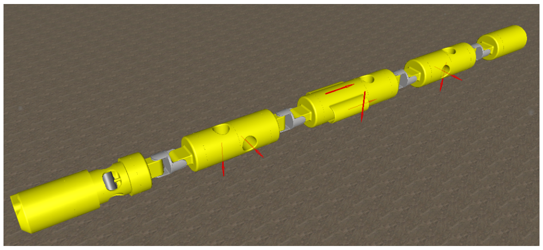


Fig. 2: Vortex model

The USM modelled in Vortex is shown in Figure 2. The model consists of five links with link lengths $l = [l_1, l_2, l_3, l_4, l_5] = [0.507, 0.688, 0.831, 0.688, 0.474]$ meters, connected by four double joints. Although the model contains a total of eight joints, only the four horizontal joints are utilized in this study, as it is limited to 2D horizontal motion. As shown in Figure 2, the USM is equipped with thrusters, which are used for controlling the overall position and orientation of the USM. The motion of the USM in Vortex is constrained to follow exactly the velocity reference signals, ζ_{ref}, calculated by the set-based SRMTP algorithm.

A. Case 1: No set-based task

In the first case, the SRMTP framework is applied without any set-based tasks. There is only one mode, and the joint and base velocity reference signals are calculated according to

$$\zeta_{\text{ref}} = J_{12}^{\dagger}\Lambda_{12}\tilde{\sigma}_{12} + N_{12}J_3^{\dagger}\Lambda_3\tilde{\sigma}_3. \tag{24}$$

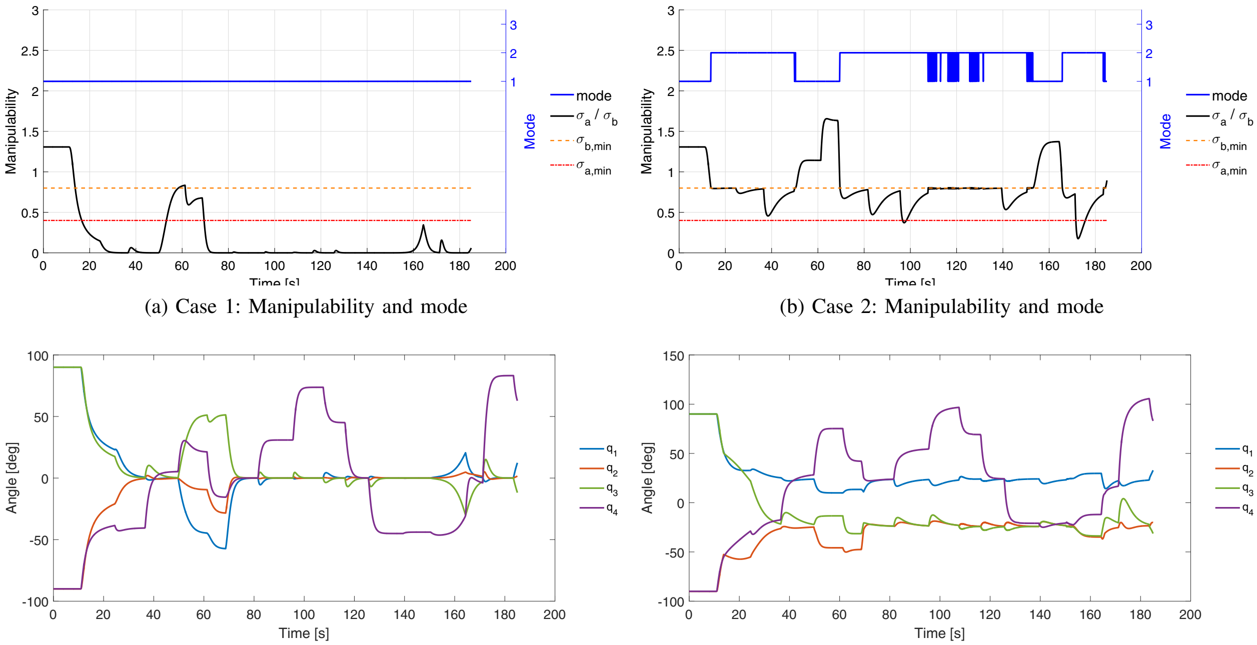
As shown in Figure 3a, without set-based tasks the manipulability is close to zero for a large part of the simulation. The USM becomes fully stretched out, and as shown in Figure 3c, the algorithm does not avoid the singular configuration associated with $q_2 = q_3 = 0$.

B. Case 2: Low-priority set-based task

In this case, we insert the set-based singularity avoidance task at the second priority level. The main reason for introducing a low-priority singularity avoidance task is to reconfigure the USM into a more dexterous shape by increasing the manipulability index. This ensures that the USM is always prepared to move the head in any direction when receiving new commands. In Mode 1, the joint and base velocity reference signals are determined as in Case 1, while in Mode 2, they are calculated according to

$$\zeta_{\text{ref}} = J_{12}^{\dagger}\Lambda_{12}\tilde{\sigma}_{12} + N_{12}J_b^{\dagger}\Lambda_b\tilde{\sigma}_b + N_{12b}J_3^{\dagger}\Lambda_3\tilde{\sigma}_3, \tag{25}$$

where the set-based task is introduced at the second priority level. Figure 3b shows that the set-based SRMTP algorithm switches from Mode 1 to Mode 2 as soon as the manipulability measure, σ_b, goes below the specified value, $\sigma_{b,\min}$. In Mode 2, the algorithm continues to pursue fulfillment of the end-effector tasks which have higher priority, while the set-based singularity avoidance task will converge towards the limit $\sigma_{b,\min}$ by a combination of changing the joint angles and moving the base of the USM. As shown in Figure 3d, the USM



(a) Case 1: Manipulability and mode

(b) Case 2: Manipulability and mode

(c) Case 1: Joint angles

(d) Case 2: Joint angles

Fig. 3: Simulation results for Case 1 and Case 2

avoids the singular configuration, $q_2 = q_3 = 0$. However, there is no guarantee that the algorithm will maintain a specified minimum distance to the singularity, since the singularity avoidance task is introduced as a low-priority set-based task. Due to the desired base position task, the USM will stretch out and linger in Mode 2 until the base is back to its initial position. Once the set-based singularity avoidance task is no longer required, the algorithm switches back to Mode 1. A practical solution to avoid chattering associated with the frequent mode switches observed in Figure 3b is to filter the output velocity reference signals. Another practical approach worth considering is the continuous task transition technique proposed in [25].

C. Case 3: High-priority set-based task

In this case, the singularity avoidance task is introduced as a set-based task with the highest priority. Figures 4a and 4b show that the manipulability measure is kept larger than or equal to the specified minimum value, $\sigma_{a,\min}$, thus ensuring avoidance of the singular configurations associated with $q_2 = q_3 = 0$. When σ_a reaches the boundary of the valid set, the set-based task is activated and the algorithm switches to Mode 2. In Mode 2 the velocity references are calculated according to

$$\zeta_{\text{ref}} = N_a J_{12}^{\dagger} \Lambda_{12} \tilde{\sigma}_{12} + N_{a12} J_3^{\dagger} \Lambda_3 \tilde{\sigma}_3, \qquad (26)$$

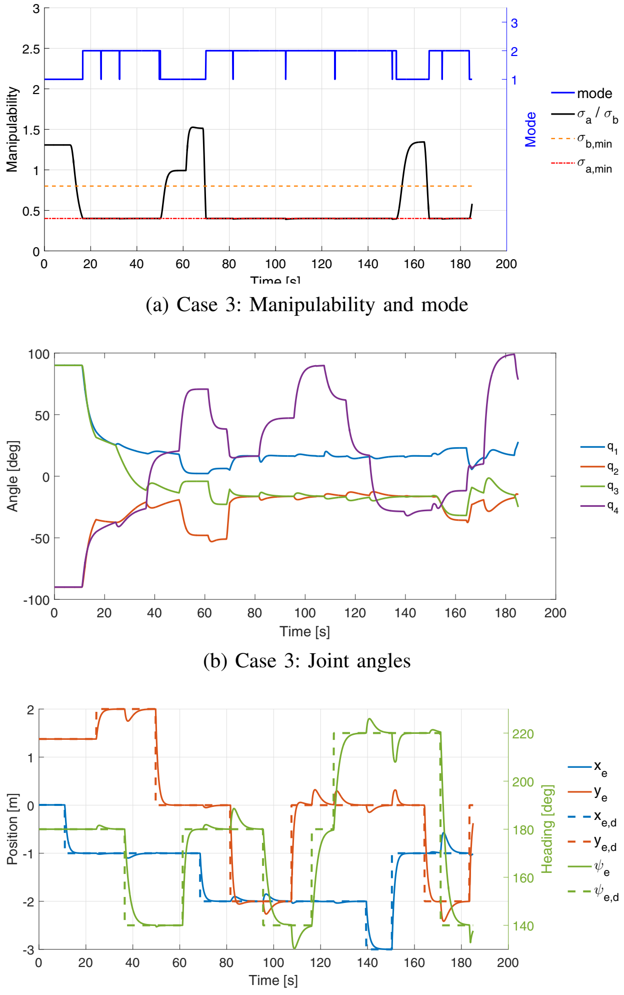
where the null space of the singularity avoidance task, N_a, ensures that σ_a is frozen at the lower boundary of the valid set and will never go below that value. Nevertheless, the task errors of the compatible end-effector equality tasks are still being satisfied, as illustrated in Figure 4c. When the end-effector task setpoints are changed, small perturbations will inevitably occur. This effect can be reduced by increasing Λ_{12}.

D. Case 4: Combined high and low priority

In the final case, we combine high and low priority set-based tasks to achieve both guaranteed singularity avoidance and to regain a dexterous configuration after a move has been performed. The two set-based task variables, σ_a and σ_b, have identical task Jacobians, $J_a = J_b$, but different desired values. As shown in Figure 5, this approach exploits the benefits of both high and low priority set-based tasks simultaneously. The manipulability measure never goes below the lower limit, $\sigma_{a,\min}$, because Mode 3 is activated at the limit. While in Mode 2, the manipulability measure converges towards the limit, $\sigma_{b,\min}$.

VI. Conclusions

This paper has proposed a novel, versatile, and widely applicable method to manage the occurrence of kinematic singularities within and at the edge of a robot's workspace. The use of the set-based SRMTP framework, and in particular the high-priority singularity avoidance task, guarantees that the robot manipulator maintains a specified minimum distance to kinematic singularities without preventing convergence of compatible equality tasks. We have demonstrated this by applying the method to a USM, and we have also shown how a low-priority set-based singularity avoidance task can be introduced to maintain a dexterous configuration. A simulation case study was carried out in order to visualize the proposed method.



(a) Case 3: Manipulability and mode

(b) Case 3: Joint angles

(c) Case 3: End-effector position and heading

Fig. 4: Simulation results for Case 3

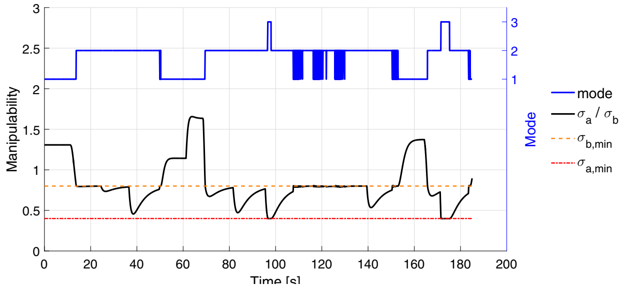


Fig. 5: Simulation results for Case 4

References

[1] T. W. McLain, "Modelling of underwater manipulator hydrodynamics with application to the coordinated control of an arm/vehicle system," Ph.D. dissertation, Stanford University, 1995.

[2] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[3] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[4] S. Chiaverini, B. Siciliano, and O. Egeland, "Kinematic analysis and singularity avoidance for a seven-joint manipulator," in *American Control Conference*, San Diego, CA, May 23-25 1990, pp. 2300–2305.

[5] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, June 1997.

[6] G. Antonelli and S. Chiaverini, "Task-priority redundancy resolution for underwater vehicle-manipulator systems," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, May 16-20 1998, pp. 768–773.

[7] ——, "Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 1, pp. 109–120, 2003.

[8] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, June 1985.

[9] S. Chiaverini, "Estimate of the two smallest singular values of the jacobian matrix: Application to damped least-squares inverse kinematics," *Journal of Robotic Systems*, vol. 10, no. 8, pp. 991–1008, 1993.

[10] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dyn. Sys., Meas., Control.*, vol. 108, no. 3, pp. 163–171, 1986.

[11] S. Chiaverini, O. Egeland, and R. K. Kanestrøm, "Achieving user-defined accuracy with damped least-squares inverse kinematics," in *Proc. 5th International Conference on Advanced Robotics (ICAR)*, Pisa, 1991, pp. 672–677.

[12] B. J. Nelson and P. K. Khosla, "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 255–269, June 1995.

[13] B. Jun, P. M. Lee, and J. Lee, "Manipulability analysis of underwater robotic arms on rov and application to task-oriented joint configuration," in *Proc. MTS/IEEE OCEANS*, Nov 2004, pp. 1548–1553.

[14] 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 International Conference on Robotics and Automation (ICRA)*, May 11-15 2002, pp. 1973–1978.

[15] J. Kim, G. Marani, W. Chung, J. Yuh, and S.-R. Oh, "Dynamic task priority approach to avoid kinematic singularity for autonomous manipulation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 30 - Oct. 4 2002, pp. 1942–1947.

[16] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, no. 16, pp. 1–18, 2016.

[17] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, Feb 2007.

[18] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, October 2009.

[19] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "The Underwater Swimming Manipulator – A Bio-Inspired AUV," in *Proc. IEEE/OES Autonomous Underwater Vehicles (AUV)*, Tokyo, Japan, Nov. 6-9 2016, pp. 387–395.

[20] E. Kelasidi, P. Liljebäck, K. Y. Pettersen, and J. T. Gravdahl, "Innovation in underwater robots: Biologically inspired swimming snake robots," *IEEE Robotics and Automation Magazine*, vol. 23, no. 1, pp. 44–62, 2016.

[21] ——, "Integral line-of-sight guidance for path following control of underwater snake robots: Theory and experiments," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 610–628, 2017.

[22] G. Antonelli, *Underwater Robots*, 3rd ed. Springer International Publishing, 2014.

[23] Vortex simulation software. CM Labs Simulations Inc. [Online]. Available: http://www.cm-labs.com/robotics

[24] Supplementary video files for IEEE CCTA 2017. Online available: https://www.dropbox.com/s/5vd33t7w7v7bpts/CCTA2017.zip.

[25] E. Simetti and G. Casalino, "A novel practical technique to integrate inequality control objectives and task transitions in priority based control," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 877–902, 2016.