

Reactive Online Motion Generation for Redundant Manipulators with Adaptive Manipulability Weighting

Ryuji Nakagawa

d5625007@aoyama.jp

Aoyama Gakuin University <https://orcid.org/0009-0001-0525-5729>

Ryosuke Tasaki

Aoyama Gakuin University

Research Article

Keywords: manipulability, redundant manipulator, adaptive weight, online motion generation, singularity avoidance

Posted Date: October 28th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11518469/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Reactive Online Motion Generation for Redundant Manipulators with Adaptive Manipulability Weighting

Ryuji Nakagawa^{1*} and Ryosuke Tasaki¹

^{1*}Department of Mechanical Engineering, Aoyama Gakuin University,
5-10-1, Fuchinobe, Chuo-ku, Sagamihara, 252-5258, Kanagawa, Japan.

*Corresponding author(s). E-mail(s): d5625007@aoyama.jp;
Contributing authors: tasaki@me.aoyama.ac.jp;

Abstract

For a manipulator to realize desired end-effector motion, it is important to maintain its kinematic capability during motion and avoid approaching singular configurations. This study proposes an online motion-generation method for point-to-point motion of redundant manipulators in obstacle-free workspaces that adjusts the priority between manipulability improvement and goal attainment according to the current state. A reference ellipsoid is updated sequentially from a path-length budget and the cumulative end-effector path length, and a manipulability reference position is generated analytically as the support point of the ellipsoid in the direction of the local manipulability gradient. An adaptive weight is then determined from the current manipulability, the predicted improvement, and the reference distance. Two task-space reference velocities, null-space posture improvement, and joint constraints are integrated into a velocity-level quadratic program. Simulations with a seven-degree-of-freedom manipulator showed that modifying the end-effector path suppressed decreases in manipulability and enabled the manipulator to reach the goal while varying the priority of improvement according to different initial states under a common set of design parameters. In transport experiments, the method generated paths at 100 Hz without a fixed reference trajectory while maintaining the end-effector orientation. Comparisons with other methods confirmed that the proposed method balanced manipulability improvement and goal attainment under the tested conditions.

Keywords: Manipulability, Redundant manipulator, Adaptive weight, Online motion generation, Singularity avoidance

1 Introduction

Manipulability is a configuration-dependent measure of a manipulator’s ability to generate end-effector velocity and is widely used to evaluate proximity to kinematic singularities [1]. Near a singularity, the available directions of motion are restricted, and large joint velocities may be required even when the target is reachable. Manipulability has therefore been considered in applications including visually guided grasping, multi-arm teleoperation, and mechanism design [2–4]. In motion planning for a redundant manipulator, it is necessary not only to satisfy the task requirements but also to maintain sufficient kinematic capability throughout the motion to realize the commanded end-effector motion.

One approach is to precompute reachability or dexterity over the workspace. End-effector positions and orientations are sampled offline and stored together with performance indices for use during planning. Capability maps and inverse reachability maps support online queries and robot placement decisions [5, 6]; they have also been used for posture selection in cooperative transport [7] and for spatial representations of dexterity [8]. However, covering a broad workspace requires robot-specific sampling and storage, and the associated cost grows with the resolution and task dimension. Moreover, because a redundant manipulator can realize the same end-effector pose with joint configurations having different manipulability values, a workspace index does not necessarily represent the inverse-kinematics solution branch currently being followed. It is therefore desirable to evaluate manipulability online from the current configuration.

Existing methods that account for manipulability can be broadly classified into control-level methods, which determine motion at each control cycle, and planning-level methods, which treat the entire path. Control-level null-space projection changes the joint configuration while preserving a prescribed end-effector motion. Consequently, it can improve manipulability only within the redundancy remaining after the primary task has been imposed and cannot modify the end-effector path itself. Quadratic-programming (QP)-based inverse kinematics can simultaneously accommodate physical constraints. Zhang et al. formulated manipulability maximization and joint constraints in a unified QP [9], while Jin et al. proposed a dynamic neural network that solves a velocity-level QP without explicitly computing an inverse matrix [10]. Dufour and Suleiman modified the path by allowing deviations from a reference trajectory, but their method requires penalties on deviations from both the reference trajectory and the reference velocity to be specified in advance [11]. Other methods have considered acceleration-level objectives and initial velocity [12], or integrated manipulability, trajectory tracking, and safety constraints into GPU-accelerated non-convex optimization [13]. NEO, proposed by Haviland and Corke, generates motion toward a target end-effector pose without requiring a predefined path. Its QP formulation allows the path to change to improve manipulability while satisfying joint constraints and avoiding obstacles [14]. Thus, control-level approaches include methods that adjust the joint configuration while preserving the prescribed end-effector motion and methods that allow the end-effector path itself to change. In the former, manipulability improvement is confined to the redundancy remaining after the primary

task has been imposed. The latter include methods that permit deviation from a reference trajectory and methods such as NEO that sequentially generate motion from the current state toward a target. Reactive generation of this kind can be effective when motion must respond to state changes during execution, as in obstacle avoidance.

At the planning level, Marić et al. optimized manipulability along a continuous-time trajectory represented by a Gaussian process [15]. Pardi et al. incorporated path length and manipulability into the cost of rapidly exploring random tree star (RRT*) [16], and Shen et al. proposed workspace planning based on closed-loop inverse kinematics and an adaptive step size [17]. These methods determine a path through trajectory optimization or search-tree construction. By contrast, in tasks such as pick-and-place, where only the start and goal positions are specified and the intermediate path can be chosen freely, online path generation can use task-space motion, in addition to null-space posture changes, to improve manipulability. Such tasks require balancing manipulability improvement, path-length limitation, and goal attainment. Their relative priorities must therefore be determined according to the current state.

The artificial potential field (APF) method is a representative reactive approach that generates a direction of motion from the current state without iterative trajectory optimization or search-tree construction [18]. The attractive effect associated with the goal can be combined with guidance obtained from an auxiliary goal, a virtual goal, or a predicted local path [19–22]. Although this structure is well suited to online motion generation, the relative strengths of multiple motion objectives must be determined. When guidance vectors conflict, an inappropriate balance can cause excessive detours, reduced resultant velocity, stagnation, or oscillation; accordingly, methods that incorporate a global path or auxiliary guidance have been proposed [23, 24]. When goal attainment is combined with manipulability improvement, their balance changes with the current state; the weights should therefore be selected based on the degree to which each objective is achieved, the resulting path length, and goal reachability.

In our previous work, we proposed a path-generation method that incorporated into a potential field guidance toward a high-manipulability end-effector position predicted from the manipulability gradient [25]. That method assigned base weights to the goal and manipulability-improvement directions in advance and prioritized the goal direction when they conflicted; it therefore did not determine the priority of manipulability improvement according to the current manipulability or the potential for improvement. The present study retains this gradient-based guidance but selects a manipulability reference position from an ellipsoidal region defined by a path-length budget. Informed RRT* uses a path-length-based ellipsoidal region to focus sampling on the subset of the state space that can yield a path shorter than the current best solution [26]. In contrast, the present study uses an ellipsoidal region that is updated during motion to generate the manipulability reference position analytically.

Based on the above discussion, this study proposes a potential-based reactive online motion-generation method that sequentially generates reference velocities in the goal and manipulability-improvement directions from the current state, without prescribing an intermediate path in advance. Specifically, a path-length budget is defined from the straight-line distance between the start and goal and an allowable detour ratio, and the remaining budget is updated by subtracting the cumulative end-effector path

length. This remaining budget defines a reference ellipsoid from which the manipulability reference position is selected analytically as the point that maximizes the manipulability predicted from the local gradient. Because this point is the support point of the ellipsoid in the gradient direction, no discrete search over candidate points is required. An adaptive weight representing the priority of manipulability improvement is determined from the current manipulability, the predicted improvement at the reference position, and the reference distance. The path-length budget, adaptive weight, and potential-based reference velocities respectively determine the admissible region for reference-position selection, the priority of manipulability improvement, and the magnitudes of the requested motions; these commands are integrated with null-space posture improvement into a velocity-level QP subject to joint-position and joint-velocity constraints. The path-length budget defines the selection region for the reference position but does not guarantee a strict upper bound on the actual path length, which is affected by discrete-time control, competition among reference velocities in the QP, and tracking errors.

The methodological contributions of this reactive online motion-generation method are as follows:

1. An analytical method for generating, without a candidate-point search, a reference position that accounts for path-length growth, based on a reference ellipsoid updated as the cumulative end-effector path length increases and on the local manipulability gradient.
2. A method for adaptively determining the priority of manipulability improvement from a relative assessment of the predicted improvement over the current value, without directly using the global maximum manipulability in the weight calculation.

The proposed method is evaluated in terms of path length, manipulability, and computation time. The scope is obstacle-free point-to-point motion; the method does not guarantee global optimality, goal attainment from an arbitrary initial state, or avoidance of local stagnation.

2 Manipulability Reference Position Generation

This section generates a task-space reference position for improving manipulability from the current state. The admissible region for the reference position is constrained by a path-length budget that is updated according to the accumulated path length.

2.1 Task-Space Manipulability Gradient

The relationship between the position and orientation of a manipulator's end-effector and its joint coordinates is described by its kinematics.

For a redundant manipulator with $n > 3$ degrees of freedom, the relationship among the translational end-effector velocity, angular end-effector velocity, and joint velocity is expressed as

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = J(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} J_v(\mathbf{q}) \\ J_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}}, \quad (1)$$

where $J_v \in \mathbb{R}^{3 \times n}$ and $J_\omega \in \mathbb{R}^{3 \times n}$ are the translational and rotational components of the Jacobian, respectively. The basic formulation below addresses point-to-point motion of the end-effector position and does not constrain the end-effector orientation as a primary task. Therefore, $J_v(t) = J_v(\mathbf{q}(t))$ is used for both manipulability evaluation and translational velocity generation, with $\dot{\mathbf{x}}(t) = J_v(t)\dot{\mathbf{q}}(t)$. In Section 5, the rotational velocity constraint $J_\omega(t)\dot{\mathbf{q}}(t) = \mathbf{0}$ is added as an application to a transport task with an orientation constraint. Nevertheless, the translational manipulability $m(t)$ defined below is used consistently to generate the reference position, adaptive weight, and motion. Translational manipulability under the orientation constraint, $m_c(t)$, is introduced in Section 5 solely to evaluate the resulting motion. Yoshikawa's translational manipulability is

$$m(t) = \sqrt{\det(J_v(t)J_v(t)^T)}. \quad (2)$$

This index is proportional to the volume of the translational manipulability ellipsoid and approaches zero as the manipulator approaches a singular configuration [1].

The joint-space gradient $\nabla_{\mathbf{q}}m(t)$ is evaluated analytically at each control cycle. Following Park et al. [27], let $H_{v,k}(t) = \partial J_v / \partial q_k$ denote the derivative of the translational Jacobian with respect to the k th joint. Each component of the gradient is then given by

$$\frac{\partial m}{\partial q_k} = m(t) \operatorname{tr} \left[\left(J_v(t)J_v(t)^T \right)^{-1} J_v(t)H_{v,k}(t)^T \right]. \quad (3)$$

This analytical expression does not require repeated finite-difference evaluations and is consistent with formulations based on the manipulator Hessian [28]. The minimum-norm least-squares solution that maps the joint-space gradient into the translational task space is defined as

$$\mathbf{g}(t) = \left(J_v(t)^T \right)^+ \nabla_{\mathbf{q}}m(t), \quad (4)$$

where $(\cdot)^+$ denotes the Moore–Penrose pseudoinverse. For a small end-effector displacement $\Delta \mathbf{x}$, the change in manipulability can be approximated as $\Delta m \simeq \mathbf{g}(t)^T \Delta \mathbf{x}$. Thus, $\mathbf{g}(t)$ is not a global mapping determined solely by the end-effector position; it is a local gradient that depends on the current configuration. Near a singular configuration, $J_v J_v^T$ becomes ill-conditioned, which may reduce the accuracy of both the gradient and the first-order prediction.

2.2 Path-Length Budget and Reference Ellipsoid

Modifying the end-effector path to improve manipulability increases the path length relative to the shortest path connecting the start and goal positions. We therefore introduce a remaining path-length budget, obtained by sequentially subtracting the accumulated path length from the path-length budget, and express it as the spatial

freedom available for selecting the manipulability reference position. Let $\mathbf{x}_s$ and $\mathbf{x}_g$ denote the start and goal positions of the end-effector, respectively. The straight-line distance between them and the path-length budget are defined as

$$\begin{aligned} L_{\min} &= \|\mathbf{x}_g - \mathbf{x}_s\|, \\ L_{\max} &= r_{\text{detour}} L_{\min}, \quad r_{\text{detour}} \geq 1. \end{aligned} \quad (5)$$

In an obstacle-free workspace, $L_{\min}$ is a lower bound on the length of an end-effector path that reaches the goal. When the allowable detour ratio r_{detour} is 1, the initial admissible region for the reference position is the line segment connecting the start and goal; increasing the ratio expands this region. $L_{\max}$ is a design quantity that defines the selection region for the reference position, whereas r_{detour} does not directly determine the priority of manipulability improvement.

The cumulative end-effector path length $L(t)$ and remaining path-length budget $B(t)$ are defined as

$$\begin{aligned} L(t) &= \sum_{\tau=0}^{t-1} \|\mathbf{x}(\tau+1) - \mathbf{x}(\tau)\|, \\ B(t) &= L_{\max} - L(t). \end{aligned} \quad (6)$$

Because the end-effector position at each instant is obtained from the joint state by forward kinematics, $L(t)$ reflects the actual end-effector trajectory generated up to that instant rather than a nominal path assumed during planning. $B(t)$ represents the remaining path-length budget allocated to the path from the current position to the goal. To select a reference position according to this budget, a candidate $\mathbf{x}$ is required to satisfy

$$L(t) + \|\mathbf{x} - \mathbf{x}(t)\| + \|\mathbf{x}_g - \mathbf{x}\| \leq L_{\max}. \quad (7)$$

The left-hand side is the sum of the accumulated path length, the straight-line distance from the current position to the candidate position, and the straight-line distance from the candidate position to the goal. Thus, this condition requires the sum of the accumulated path length and the length of the path that proceeds directly to the goal through the candidate position to be no greater than $L_{\max}$. Because the set of points whose summed distances from two fixed points are no greater than a constant forms an ellipsoid, the admissible region is a three-dimensional ellipsoid defined by the current position $\mathbf{x}(t)$, the goal position $\mathbf{x}_g$, and the full major-axis length $B(t)$. This region is referred to as the reference ellipsoid. Converting the path-length budget into a spatial region allows the admissible region for the reference position to be treated directly in task space.

Let $D(t) = \|\mathbf{x}_g - \mathbf{x}(t)\|$, $\mathbf{c}(t) = (\mathbf{x}(t) + \mathbf{x}_g)/2$, and $\mathbf{e}(t) = (\mathbf{x}_g - \mathbf{x}(t))/D(t)$, and define the major and minor semiaxis lengths as $a(t) = B(t)/2$ and $b(t) = \sqrt{B(t)^2 - D(t)^2}/2$, respectively. The shape matrix is

$$P(t) = b(t)^2 I + (a(t)^2 - b(t)^2) \mathbf{e}(t) \mathbf{e}(t)^T. \quad (8)$$

The eigenvalue of $P(t)$ associated with the goal direction is $a(t)^2$, and the eigenvalues in the two orthogonal directions are $b(t)^2$. The ellipsoid aligned with

the goal direction can therefore be represented without explicitly constructing a coordinate-dependent rotation matrix. The admissible region is expressed as

$$(\mathbf{x} - \mathbf{c}(t))^T P(t)^{-1} (\mathbf{x} - \mathbf{c}(t)) \leq 1. \quad (9)$$

Because $B(t)$ decreases as $L(t)$ increases, the reference ellipsoid progressively contracts toward the line segment connecting the current and goal positions. Early in the motion, when ample path-length budget remains, the reference position can be selected from a broad region. Later, after the budget has been consumed, the admissible region approaches the goal direction. A nondegenerate ellipsoid is constructed only when $B(t) > D(t) > 0$. This condition, however, constrains only the length of the geometric path through the reference position. Because the QP described later simultaneously accounts for the goal direction, manipulability improvement, and joint constraints, the actual end-effector does not follow that geometric path exactly. The reference ellipsoid is therefore a geometric measure for limiting path length and does not strictly guarantee $L(t) \leq L_{\max}$ for the actual path, which is affected by discrete-time control, competition between the reference velocities, and tracking errors.

2.3 Analytical Computation of the Reference Position

The task-space gradient $\mathbf{g}(t)$ provides the local end-effector direction predicted to yield the greatest increase in manipulability at the current configuration, but it does not specify how far to move in that direction. This section therefore combines this local information with the reference ellipsoid to determine the reference position that maximizes predicted manipulability within the region permitted by the remaining path-length budget. The manipulability at an arbitrary position $\mathbf{x}$ in the ellipsoid is predicted using the first-order model about the current position,

$$\hat{m}(\mathbf{x}) = m(t) + \mathbf{g}(t)^T (\mathbf{x} - \mathbf{x}(t)). \quad (10)$$

This model corresponds to a first-order Taylor expansion of manipulability, and $\mathbf{g}(t)^T (\mathbf{x} - \mathbf{x}(t))$ represents the predicted change in manipulability due to displacement from the current position. Because manipulability is a nonlinear index that depends on the joint configuration, this approximation is used as a local model around the current state. Terms independent of $\mathbf{x}$ do not affect the solution; hence, the reference position that maximizes predicted manipulability is the solution to

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{g}(t)^T \mathbf{x} \\ \text{s.t.} \quad & (\mathbf{x} - \mathbf{c}(t))^T P(t)^{-1} (\mathbf{x} - \mathbf{c}(t)) \leq 1. \end{aligned} \quad (11)$$

Because the objective is linear and the feasible region is a convex ellipsoid, the global optimum of this first-order prediction problem can be obtained analytically. However, the resulting point is not the point of globally maximum manipulability over the entire workspace; rather, under the local gradient obtained from the current configuration, it represents the best direction within the remaining path-length budget. Transforming the ellipsoid into a unit sphere by $\mathbf{y} = P(t)^{-1/2} (\mathbf{x} - \mathbf{c}(t))$ shows, through the Cauchy–Schwarz inequality, that the maximum is attained in the direction of the

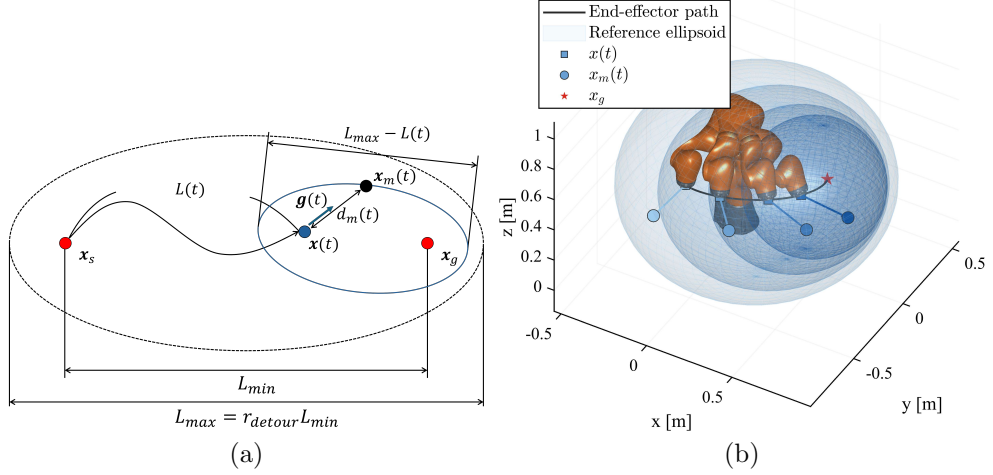


Fig. 1 Reference ellipsoid and manipulability reference position. (a) Conceptual illustration in which the initial and progressively contracted reference ellipsoids are superimposed in the same plane. (b) Example evolution of the reference ellipsoid during motion, showing the generated end-effector path, reference ellipsoids at regular intervals, current position $\mathbf{x}(t)$, manipulability reference position $\mathbf{x}_m(t)$, and goal position $\mathbf{x}_g$.

transformed gradient. Transforming back to the original coordinates yields the support point of the ellipsoid in the gradient direction:

$$\mathbf{x}_m(t) = \mathbf{c}(t) + \frac{P(t)\mathbf{g}(t)}{\sqrt{\mathbf{g}(t)^T P(t)\mathbf{g}(t)}}. \quad (12)$$

For $\mathbf{g}(t) \neq \mathbf{0}$, this point lies on the boundary of the ellipsoid. Because the shape matrix transforms the gradient direction according to the ellipsoid axes, and thus according to the remaining geometric freedom, the reference position is not simply a point located a fixed distance along the gradient direction. It can be computed uniquely through a fixed sequence of operations, without a discrete search over candidate points or iterative optimization, making it suitable for online motion generation in which the reference region changes at every control cycle. The predicted value and reference distance used in the next section are

$$\begin{aligned} m_{\text{ref}}(t) &= m(t) + \mathbf{g}(t)^T (\mathbf{x}_m(t) - \mathbf{x}(t)), \\ d_m(t) &= \|\mathbf{x}_m(t) - \mathbf{x}(t)\|. \end{aligned} \quad (13)$$

$m_{\text{ref}}(t)$ is not the true manipulability guaranteed after moving to the reference position; it is a predicted quantity used to determine the priority of improvement from the local gradient. In contrast, $d_m(t)$ represents the spatial magnitude of the requested motion in the reference direction and is reflected in the distance-dependent reference velocity defined in Section 3.2. Figure 1(a) illustrates the concept of the reference ellipsoid and the manipulability reference position, and Figure 1(b) shows a specific evolution of the reference ellipsoid during motion. In Figure 1(b), as the generated path extends, the remaining path-length budget decreases, the reference ellipsoids shown at regular intervals progressively contract, and the manipulability reference position

approaches the current position. Consequently, the reference distance $d_m(t)$ decreases, weakening the velocity request in the manipulability-improvement direction during the latter part of the motion.

3 Adaptive Online Motion Generation

This section converts the manipulability reference into a state-dependent task-space command and integrates it with goal-directed motion, null-space posture improvement, and joint constraints.

3.1 Adaptive Manipulability Weight

Conventional optimization methods that incorporate manipulability into the objective require its weight to be specified in advance and may require scaling for the motion under consideration. Moreover, a constant manipulability weight may cause unnecessary path modifications when manipulability is already sufficient, while providing inadequate guidance when the current value is low and substantial improvement is possible. We therefore determine a tracking weight from the predicted improvement at the reference position according to the current kinematic state. The predicted improvement is defined as

$$\Delta m(t) = m_{\text{ref}}(t) - m(t). \quad (14)$$

If $\Delta m(t)$ were used directly as the weight, the predicted improvement would increase with the distance to the reference position even for the same local gradient direction, making the weight dependent on the size of the reference ellipsoid and the allowable detour ratio. We therefore normalize the predicted improvement by the reference distance $d_m(t)$, thereby removing its direct dependence on the distance to the reference position, and then divide it by the current manipulability $m(t)$ to obtain the relative potential for improvement from the current state. Because this quantity has dimensions of inverse length, it is multiplied by a representative distance d_0 to make it dimensionless. To prevent division by zero, small constants $\varepsilon_d = 10^{-3}$ m and $\varepsilon_m = 10^{-6}$ are added to the reference distance and current manipulability, respectively. ε_m is treated in the same unit system as manipulability. The adaptive weight, with negative predicted improvements set to zero and an upper bound w_{max} , is defined as

$$w_m(t) = \min \left\{ w_{\text{max}}, \frac{d_0 \max(0, \Delta m(t))}{(d_m(t) + \varepsilon_d)(m(t) + \varepsilon_m)} \right\}. \quad (15)$$

Scenario C in Section 4.2, which has the lowest initial manipulability among the scenarios evaluated, was used as the reference state for determining the scale of the adaptive weight. Substituting its initial configuration into Eq. (15) gives the condition $d_0 \leq 0.567$ m for the weight before clipping not to exceed $w_{\text{max}} = 3.0$. We therefore set $d_0 = 0.5$ m to avoid saturation at the upper bound from the start of the motion and to retain weight variation as the state changes. In the QP in Section 3.3, the weight on the tracking term for the goal reference velocity is 1. Relative to this baseline, the upper bound is set to $w_{\text{max}} = 3.0$ to prevent the effect of the tracking term for the manipulability reference velocity from becoming excessive. This upper bound is not

an experimentally optimal value, nor does it guarantee that the goal direction always takes priority. d_0 sets the scale of the adaptive weight, whereas $w_{\max}$ sets its upper bound.

This weight is a measure of the relative predicted improvement normalized by the distance to the reference position. Because the predicted improvement is obtained from a first-order model, normalization by the reference distance suppresses the scale effect caused solely by the remoteness of the reference position and extracts a local improvement trend corresponding to the directional derivative along the reference direction. Near a singularity, $w_m(t)$ becomes large when improvement is possible relative to the small current manipulability. Near a local maximum, both the gradient and predicted improvement become small, and the weight therefore also becomes small. This mechanism adjusts the balance between goal attainment and manipulability improvement from the kinematic state at each control cycle rather than fixing it before the motion. When no reference position is generated, the gradient is zero, or the calculated weight is nonfinite, $w_m(t)$ is set to zero.

The path-length budget, adaptive weight, and reference velocity determine the admissible region for the reference position, the priority of manipulability improvement, and the magnitude of the requested motion, respectively.

3.2 Potential-Based Reference Velocities

This section generates motion commands in the goal and manipulability-reference directions using a bounded distance-dependent law corresponding to an attractive potential. The same basic velocity law is used for both directions, with additional attenuation applied to the manipulability reference velocity near the goal. Define the goal distance as $d_g(t) = \|\mathbf{x}_g - \mathbf{x}(t)\|$ and, for $d_g(t) > 0$, the goal direction as $\mathbf{u}_g(t) = (\mathbf{x}_g - \mathbf{x}(t))/d_g(t)$. The goal reference velocity is

$$\mathbf{v}_g(t) = \alpha_g \left[1 - \left(\frac{\beta_g}{d_g(t) + \beta_g} \right)^2 \right] \mathbf{u}_g(t). \quad (16)$$

α_g is the nominal speed, and β_g is the distance scale of the velocity variation. This velocity points toward the goal and approaches the nominal speed far from the goal, while continuously decaying to zero as the end-effector approaches the goal.

For $d_m(t) > 0$, define the manipulability reference direction as $\mathbf{u}_m(t) = (\mathbf{x}_m(t) - \mathbf{x}(t))/d_m(t)$. The terminal attenuation factor based on the goal distance is defined as

$$\gamma(t) = \frac{d_g(t)^2}{d_g(t)^2 + d_c^2}. \quad (17)$$

Multiplying the basic distance-dependent velocity by this factor gives the manipulability reference velocity,

$$\mathbf{v}_m(t) = \alpha_m \gamma(t) \left[1 - \left(\frac{\beta_m}{d_m(t) + \beta_m} \right)^2 \right] \mathbf{u}_m(t). \quad (18)$$

In this study, $\alpha_m = \alpha_g = 0.5$ m/s and $\beta_m = \beta_g = 0.3$ m, and the terminal attenuation distance is set to $d_c = \beta_m = 0.3$ m.

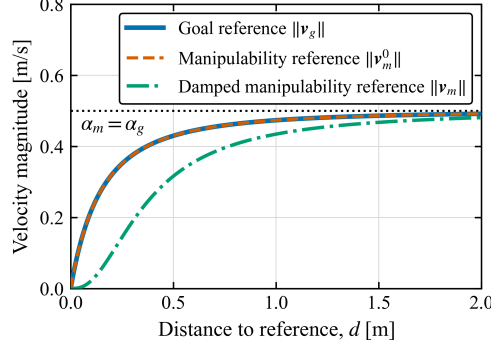


Fig. 2 Distance characteristics of the goal reference velocity $\|\mathbf{v}_g\|$, the undamped manipulability reference velocity $\|\mathbf{v}_m^0\|$, and the damped manipulability reference velocity $\|\mathbf{v}_m\|$. The horizontal axis is the distance to the corresponding goal or reference position, and the vertical axis is the velocity magnitude. The figure uses $d_g = d_m = d$ for illustration and does not assume that the two distances are always equal during the actual motion. The two undamped curves coincide, while damping further suppresses the velocity command in the manipulability direction near the goal.

The manipulability reference velocity decays to zero as the end-effector approaches the reference position. In addition, the terminal attenuation factor $\gamma(t)$ is zero at the goal, equals $1/2$ when $d_g(t) = d_c$, and approaches 1 far from the goal. Thus, even when unused path-length budget remains and the manipulability reference position is distant from the current position, the command in the manipulability direction weakens as the end-effector approaches the goal. For a fixed goal distance, the magnitude of the manipulability reference velocity approaches $\alpha_m \gamma(t)$ as the reference distance increases. Terminal attenuation is applied only to the manipulability reference velocity and is not applied again to the adaptive weight.

Figure 2 shows the distance characteristics of the goal reference velocity and the manipulability reference velocity before and after terminal attenuation. The manipulability reference velocity before attenuation is denoted by $\mathbf{v}_m^0$.

When $B(t) \leq D(t)$ or $\mathbf{g}(t) = \mathbf{0}$, no manipulability reference is generated, and $w_m(t) = 0$ and $\mathbf{v}_m(t) = \mathbf{0}$. To avoid division by zero when computing direction vectors, the manipulability reference velocity is defined as zero when $d_m(t) = 0$, and both reference velocities are defined as zero when $d_g(t) = 0$.

3.3 Joint-Constrained Velocity Optimization

This section converts the two task-space reference velocities into a feasible joint velocity while using the redundant degrees of freedom to improve manipulability. In addition to modifying the task-space path, the null space of $J_v(t)$ is used to improve the joint configuration. The null-space projection matrix and reference joint velocity are defined as

$$\begin{aligned} N(t) &= I - J_v(t)^+ J_v(t), \\ \dot{\mathbf{q}}_n^{\text{ref}}(t) &= k_n N(t) \nabla_q m(t). \end{aligned} \tag{19}$$

Because $J_v(t)N(t) = \mathbf{0}$, the null-space reference improves manipulability without changing the translational motion of the end-effector to first order.

The final joint velocity is obtained by solving

$$\begin{aligned}
\min_{\dot{\mathbf{q}}(t)} \quad & \|J_v(t)\dot{\mathbf{q}}(t) - \mathbf{v}_g(t)\|^2 \\
& + w_m(t)\|J_v(t)\dot{\mathbf{q}}(t) - \mathbf{v}_m(t)\|^2 \\
& + w_n\|N(t)\dot{\mathbf{q}}(t) - \dot{\mathbf{q}}_n^{\text{ref}}(t)\|^2 \\
& + \lambda_q\|\dot{\mathbf{q}}(t)\|^2 \\
\text{s.t.} \quad & \mathbf{l}(t) \leq \dot{\mathbf{q}}(t) \leq \mathbf{u}(t).
\end{aligned} \tag{20}$$

The first term requests progress toward the goal, and the second introduces task-space motion toward the manipulability reference. When the two directions conflict, $w_m(t)$ determines their state-dependent balance. The empirically selected values are $k_n = 10$ and $w_n = 1$. The third term tracks the null-space reference, whereas the fourth regularizes the joint velocity with $\lambda_q = 0.1$ to suppress unnecessarily large velocity commands [29]. Because $\lambda_q > 0$ and the upper and lower constraints are linear, this problem is a convex QP.

Let $\mathbf{q}_{\min}$ and $\mathbf{q}_{\max}$ denote the lower and upper joint-position limits, $\dot{\mathbf{q}}_{\max}$ the absolute joint-velocity limits, and Δt the control period. The elementwise lower and upper bounds that simultaneously satisfy the joint-position and joint-velocity limits are

$$\begin{aligned}
\mathbf{l}(t) &= \max\left(-\dot{\mathbf{q}}_{\max}, \frac{\mathbf{q}_{\min} - \mathbf{q}(t)}{\Delta t}\right), \\
\mathbf{u}(t) &= \min\left(\dot{\mathbf{q}}_{\max}, \frac{\mathbf{q}_{\max} - \mathbf{q}(t)}{\Delta t}\right).
\end{aligned} \tag{21}$$

Both reference velocities decay to zero near the goal, and the additional attenuation of the manipulability reference velocity suppresses residual improvement commands. However, because the generated velocity may decrease when the two task-space directions conflict, the method does not guarantee avoidance of local stagnation or global goal attainment from an arbitrary initial state.

3.4 Online Motion Generation Algorithm

Figure 3 shows the structure of the proposed framework, and Algorithm 1 summarizes the computation at each control cycle. The reference ellipsoid, manipulability reference, and adaptive weight are updated from the measured joint state and cumulative end-effector path length.

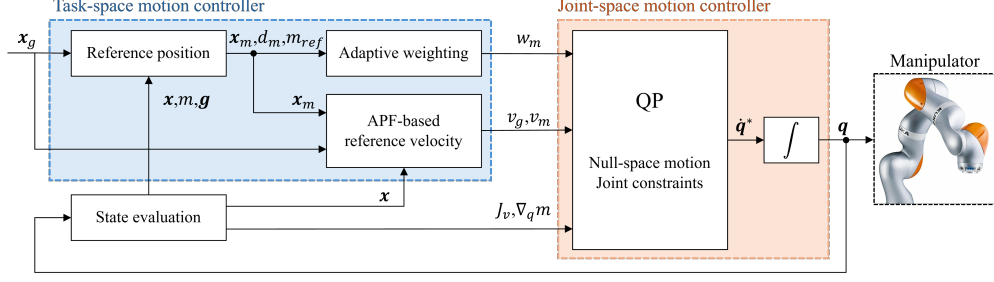


Fig. 3 Proposed online motion-generation framework. The QP combines the goal and manipulability reference velocities with null-space posture improvement, using an adaptive weight for the manipulability objective and enforcing joint-position and joint-velocity limits.

Algorithm 1 Online motion generation based on an adaptive manipulability weight

Input: $q(0)$, x_g , r_{detour} , Δt , $\varepsilon_{\text{goal}}$, T_{max}

Output: $\dot{q}^*(t)$

Initialization:

$x_s \leftarrow \text{ForwardKinematics}(q(0))$, $L(0) \leftarrow 0$, $L_{\min} \leftarrow \|x_g - x_s\|$, $L_{\max} \leftarrow r_{\text{detour}} L_{\min}$

for each control cycle t **do**

$q(t) \leftarrow \text{MeasureJointPosition}()$

$(x(t), J(t), J_v(t)) \leftarrow \text{Kinematics}(q(t))$

if $t > 0$ **then**

$L(t) \leftarrow L(t-1) + \|x(t) - x(t-1)\|$

end if

$D(t) \leftarrow \|x_g - x(t)\|$

if $D(t) \leq \varepsilon_{\text{goal}}$ or $t\Delta t \geq T_{\text{max}}$ **then**

$\dot{q}^*(t) \leftarrow \mathbf{0}$

 Output $\dot{q}^*(t)$ and terminate the iteration

end if

$(m(t), \nabla_q m(t), g(t)) \leftarrow \text{ManipulabilityState}(q(t), J_v(t))$

$B(t) \leftarrow L_{\max} - L(t)$

$(w_m(t), v_m(t)) \leftarrow (0, \mathbf{0})$

if $B(t) > D(t) > 0$ and $\|g(t)\| > 0$ **then**

$\mathcal{E}(t) \leftarrow \text{ReferenceEllipsoid}(x(t), x_g, B(t))$

$(x_m(t), m_{\text{ref}}(t), d_m(t)) \leftarrow \text{ManipulabilityReference}(\mathcal{E}(t), g(t), m(t))$

$w_m(t) \leftarrow \text{AdaptiveWeight}(m(t), m_{\text{ref}}(t), d_m(t))$

$v_m(t) \leftarrow \text{ManipulabilityVelocity}(x(t), x_m(t), x_g)$

end if

$v_g(t) \leftarrow \text{GoalVelocity}(x(t), x_g)$

$(N(t), \dot{q}_n^{\text{ref}}(t)) \leftarrow \text{NullSpaceReference}(q(t), J_v(t))$

$(l(t), u(t)) \leftarrow \text{JointVelocityBounds}(q(t), \Delta t)$

$\dot{q}^*(t) \leftarrow \text{SolveQP}(t)$

 Output $\dot{q}^*(t)$

end for

In simulation, the joint state is updated as $\mathbf{q}(t+1) = \mathbf{q}(t) + \Delta t \dot{\mathbf{q}}^*(t)$. In the real-robot implementation, the joint state measured at each control cycle is used in the next calculation. The joint-velocity command is updated at 100 Hz. The goal is considered reached when the distance to the goal position is no greater than $\varepsilon_{\text{goal}} = 0.01$ m; the motion is terminated when this condition is satisfied or the elapsed time reaches $T_{\text{max}} = 30$ s.

4 Simulation Results and Discussion

This section evaluates the design parameters and adaptive mechanism of the proposed method through simulation. First, the allowable detour ratio r_{detour} is selected from the trade-off between manipulability improvement and path length. The adaptive weight based on the normalized predicted improvement is then compared with multiple fixed weights across different motion scenarios.

The manipulator used in the simulations was a KUKA LBR iiwa 7 R800, and the simulations were performed in MATLAB 2024a. The control period was $\Delta t = 0.01$ s, and the joint state was updated at 100 Hz. The joint-position and joint-velocity limits were set to the specifications published by KUKA [30].

4.1 Allowable Detour Ratio Analysis

This section evaluates the effects of the allowable detour ratio on the actual path length and manipulability and selects the value used in the subsequent evaluations based on their trade-off.

Scenario A was used to select the allowable detour ratio. The initial joint configuration was $\mathbf{q}(0) = [-1.800, 0.000, 0.000, -1.900, 0.000, 1.100, 0.000]^T$ rad, the initial end-effector position was $\mathbf{x}_s = [-0.090, -0.386, 0.486]^T$ m, and the goal position was $\mathbf{x}_g = [0.510, 0.014, 0.496]^T$ m, giving a straight-line distance of $L_{\text{min}} = 0.721$ m. A numerical search based on iterative updates along the joint-space gradient from multiple initial configurations produced an approximate maximum manipulability of $m_{\text{max}} = 0.189$. Scenario A was configured so that its initial manipulability, $m(0) = 0.099$, was approximately half this value. Thus, its initial configuration was neither highly manipulable nor close to a singularity.

Motion-planning simulations were conducted for Scenario A with

$$r_{\text{detour}} \in \{1.5, 2.0, 2.5, 3.0\}. \quad (22)$$

The reference-velocity parameters, adaptive-weight parameters, null-space term, and QP coefficients other than the allowable detour ratio were common to all simulations. To verify the effect of modifying the end-effector path in task space, a motion with the tracking weight of the manipulability reference velocity set to $w_m = 0$ was also generated. This setting disabled only the task-space manipulability-improvement term while retaining goal tracking and null-space posture improvement. For each setting, we recorded whether the goal was reached, the actual path length L , the actual path-length ratio L/L_{min} , and the mean, minimum, and maximum manipulability along the path. Mean manipulability was defined as

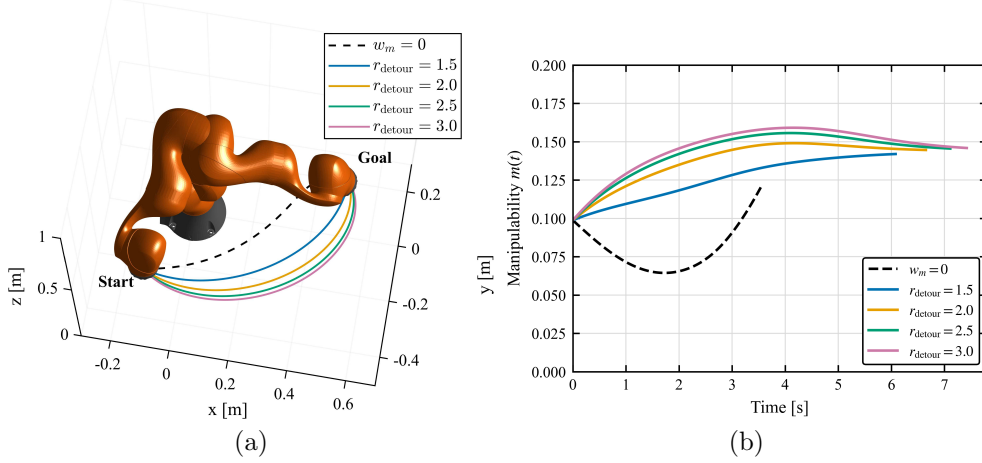


Fig. 4 Effect of the allowable detour ratio in Scenario A: (a) end-effector paths and (b) corresponding time histories of manipulability. The baseline with $w_m = 0$ is included for comparison.

$$\bar{m} = \frac{1}{N+1} \sum_{k=0}^N m(k), \quad (23)$$

where N is the number of control steps until termination.

Figure 4 shows the three-dimensional end-effector paths generated for Scenario A at each allowable detour ratio and the corresponding manipulability histories, together with the reference results for $w_m = 0$. With $w_m = 0$, the end-effector path passed close to the robot, and manipulability fell below its initial value during the middle portion of the motion. By contrast, the proposed method modified the end-effector path in the manipulability-improvement direction through the manipulability reference velocity and suppressed the decrease in manipulability at every allowable detour ratio. Compared with performing only null-space posture improvement along a goal-directed path, this result demonstrates that path modification exploiting the dependence of manipulability on end-effector position is effective in suppressing a decrease in manipulability along the path.

As the allowable detour ratio increased, the end-effector followed a more curved path and manipulability tended to improve. Under all conditions using the proposed method, the minimum manipulability occurred in the initial state, indicating that a decrease in manipulability during the motion was prevented. A larger allowable detour ratio increased the reference distance and strengthened $\mathbf{v}_m(t)$; a particularly large improvement was obtained when r_{detour} was increased from 1.5 to 2.0. For $r_{\text{detour}} \geq 2.5$, however, the changes in both the path and manipulability became smaller, indicating diminishing benefits from additional detour allowance.

Table 1 reports the specified allowable detour ratio r_{detour} together with the resulting actual path-length ratio $L/L_{\min}$.

The first row corresponds to $w_m = 0$; in this case, the manipulability reference position and allowable detour ratio do not contribute to task-space velocity generation. The goal position was reached under every condition.

Table 1 Simulation results for different allowable detour ratios and the baseline with $w_m = 0$ in Scenario A.

r_{detour}	L [m]	$L/L_{\min}$	Mean m	Minimum m	Maximum m
$w_m = 0$	0.748	1.04	0.080	0.064	0.120
1.5	0.829	1.15	0.126	0.099	0.142
2.0	0.907	1.26	0.138	0.099	0.149
2.5	0.952	1.32	0.143	0.099	0.156
3.0	0.978	1.36	0.146	0.099	0.159

Table 2 Initial conditions and goal positions for simulation Scenarios B and C.

Scenario	$q(0)$ [rad]	x_s [m]	x_g [m]	$L_{\min}$ [m]	$m(0)$
B	$[-1.800, 1.200, 0.500, -0.850, 1.000, 1.100, 0.000]^T$	$[0.097, -0.759, 0.306]^T$	$[0.497, -0.059, 0.306]^T$	0.806	0.150
C	$[-2.000, 0.800, 1.100, -1.900, 0.500, 1.600, 0.000]^T$	$[0.221, -0.310, 0.385]^T$	$[-0.080, 0.390, 0.385]^T$	0.761	0.053

The actual path-length ratio did not increase in proportion to r_{detour} because the allowable detour ratio defines the region from which the reference position can be selected rather than directly determining the strength of motion in the manipulability-improvement direction. Increasing r_{detour} from 2.5 to 3.0 produced only a small additional improvement in manipulability while increasing the actual path length. Therefore, $r_{\text{detour}} = 2.5$ was selected for the subsequent evaluations based on the trade-off between manipulability improvement and path length.

4.2 Evaluation of Adaptive Manipulability Weighting

This section verifies whether the adaptive weight based on the normalized predicted improvement can change the priority of manipulability improvement according to the motion state. The value $r_{\text{detour}} = 2.5$, selected from Scenario A in Section 4.1, was used for all scenarios. In addition to Scenario A, Scenario B was configured with high initial manipulability and a long distance to the goal. Scenario C was configured with low initial manipulability and a strong need for improvement while moving toward the goal. Table 2 lists Scenarios B and C.

Five settings were compared: the proposed adaptive weight and fixed weights $w_m = 0.5, 1.5, 2.5$, and 3.5 . The same settings were used across all scenarios. Because the adaptive weight satisfies $0 \leq w_m(t) \leq 3.0$, the fixed weights ranged from a low value to 3.5 , which exceeds this upper bound. Reference-position generation, reference velocities, the null-space term, QP coefficients, and joint constraints were identical across all settings except for the weight.

Figure 5 summarizes the results for Scenarios A, B, and C, with the end-effector paths in the left column and the corresponding manipulability histories in the right column. Although a larger fixed weight strengthens the tracking command for manipulability improvement, it also changes path length and reachability; consequently, the mean and minimum manipulability do not improve monotonically. With the fixed weight $w_m = 0.5$, the minimum manipulability values in Scenarios A and C were 0.086 and 0.037, respectively, both below their initial values. With the adaptive weight, the

respective minima were maintained at the initial values of 0.099 and 0.053, suppressing these decreases. In Scenario B, which had high initial manipulability, the actual path-length ratio with the adaptive weight was 1.11, remaining close to the value of 1.09 for the fixed weight $w_m = 0.5$, while both the mean and minimum manipulability were higher. However, the goal-reaching time was shorter with the fixed weight $w_m = 0.5$. In Scenario C, which had low initial manipulability, a large detour occurred immediately after motion onset, producing motion that preferentially compensated for the manipulability deficit. These results are consistent with the intended behavior of the adaptive weight: limiting path modification when initial manipulability is high and prioritizing improvement when it is low. Figure 6 shows the temporal variation in the adaptive weight for each scenario.

Table 3 summarizes the quantitative results for all scenarios. The time column gives the time required to reach the goal, and a dash (–) denotes a condition that failed to reach the goal and was forcibly terminated at 30 s. For the corresponding condition—Scenario B with the fixed weight $w_m = 1.5$ —the path-length ratio and the mean and minimum manipulability were calculated from the data up to forced termination.

In Scenario A, the mean manipulability with the adaptive weight was lower than that with the fixed weight $w_m = 1.5$, but both the actual path-length ratio and goal-reaching time were smaller. In Scenario C, the adaptive weight produced a longer path than $w_m = 1.5$ and thereby achieved higher mean manipulability, while reaching the goal with a shorter path and in less time than $w_m = 2.5$ and 3.5. In Scenario B, $w_m = 1.5$ yielded high mean manipulability but failed to reach the goal before termination at 30 s. By contrast, the proposed method with the adaptive weight reached the goal while suppressing a decrease in manipulability and avoiding unnecessary detours in the high-initial-manipulability scenario; in the low-initial-manipulability scenario, it sufficiently improved manipulability and ultimately reached the goal. Although the adaptive weight did not outperform every fixed weight for every metric, it changed the priority of improvement according to the state while using common design parameters across all scenarios, thereby reducing reliance on scenario-specific tuning of a fixed manipulability weight.

5 Transport Experiments with Fixed End-Effector Orientation

This section applies the proposed method to a transport task in which a grasped object is moved from a start position to a goal position while maintaining its orientation. Because the orientation of an object may be constrained when transporting liquids or precision equipment, a glass containing liquid was used as an example of a transport task with an orientation constraint. The purpose, however, was not vibration suppression for liquid transport; neither the amplitude of free-surface oscillations nor vibration-suppression performance was evaluated. In addition to a standard transport scenario, a scenario in which following a straight-line reference trajectory tends to bring the manipulator close to a singular configuration was established to compare the proposed and conventional methods. The generated path was evaluated in terms

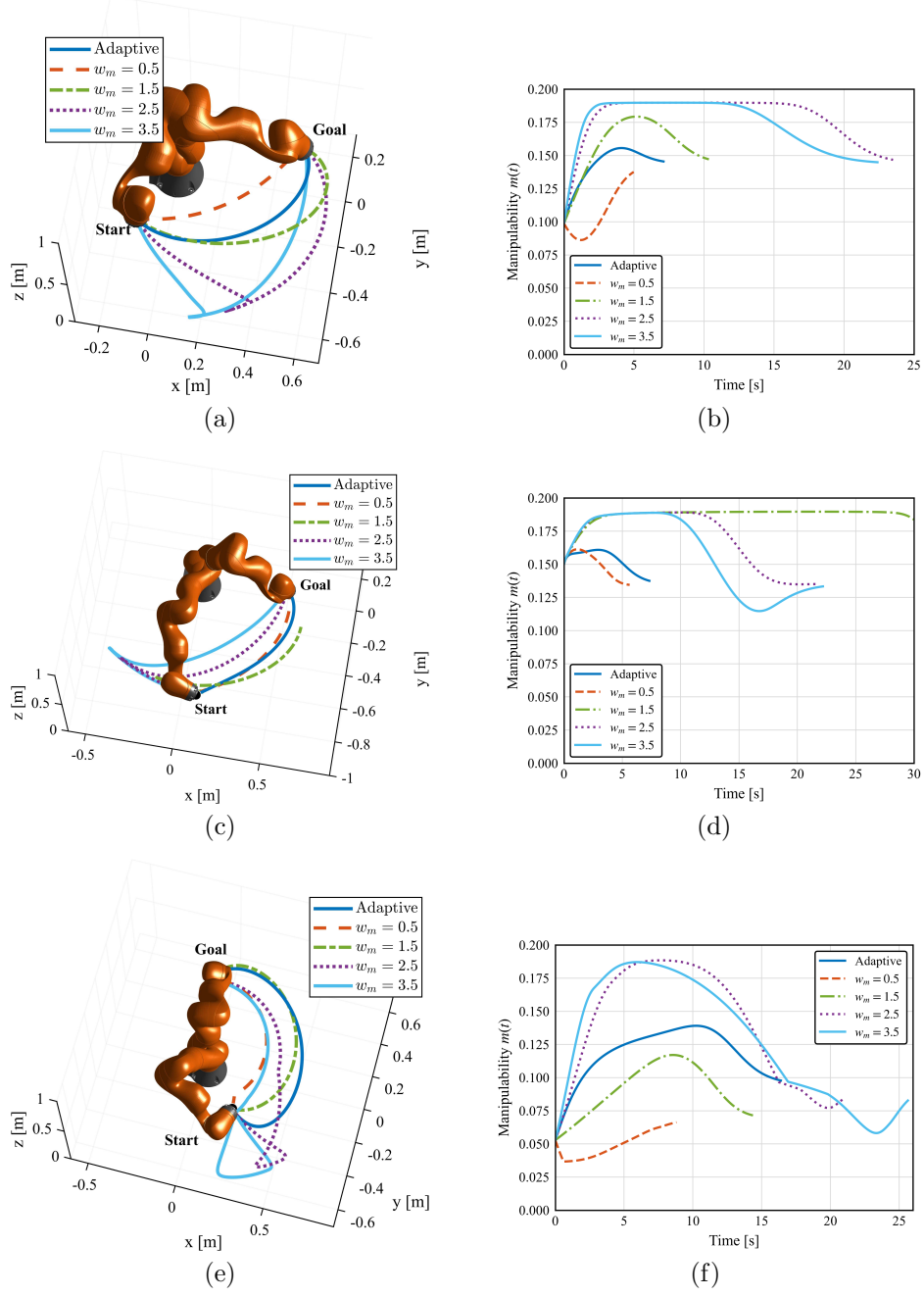


Fig. 5 Comparison of the adaptive and fixed weights in Scenarios A, B, and C. Panels (a), (c), and (e) show three-dimensional end-effector paths, and panels (b), (d), and (f) show the corresponding time histories of manipulability. Panels (a) and (b), (c) and (d), and (e) and (f) correspond to Scenarios A, B, and C, respectively.

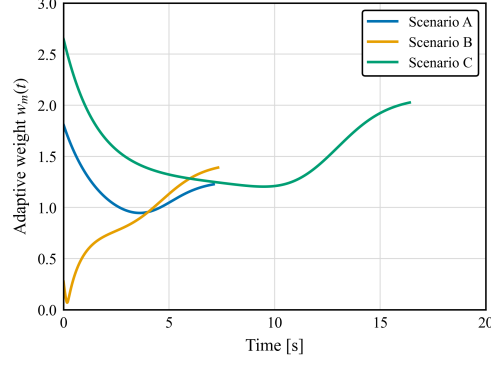


Fig. 6 Time histories of the adaptive weight $w_m(t)$ for Scenarios A, B, and C with the proposed method.

Table 3 Comparison of adaptive and fixed manipulability weights.

Scenario	w_m	$L/L_{\min}$	Mean m	Minimum m	Time [s]
A	Adaptive	1.32	0.143	0.099	7.12
A	0.5	1.04	0.106	0.086	4.99
A	1.5	1.51	0.158	0.099	10.36
A	2.5	2.43	0.178	0.099	23.51
A	3.5	2.48	0.174	0.099	22.40
B	Adaptive	1.11	0.153	0.138	7.33
B	0.5	1.09	0.150	0.134	5.64
B	1.5	1.98	0.187	0.150	—
B	2.5	2.48	0.170	0.135	21.82
B	3.5	2.48	0.158	0.115	22.20
C	Adaptive	2.07	0.118	0.053	16.41
C	0.5	1.28	0.050	0.037	8.82
C	1.5	1.89	0.091	0.053	14.38
C	2.5	2.55	0.140	0.053	21.04
C	3.5	2.71	0.128	0.053	25.64

of manipulability, end-effector orientation error, and computation time to verify the path-generation capability and real-time feasibility of the proposed method.

5.1 Experimental Setup and Evaluation Metrics

Experiments were conducted using a KUKA LBR iiwa 7 R800 seven-degree-of-freedom manipulator, and the proposed and comparison methods were implemented in the Robot Operating System (ROS). The control system used ROS 2 Humble with an Intel Core i7-13620H processor. At each control cycle, the current joint position $\mathbf{q}(t)$ was obtained from the robot, and the joint velocity $\dot{\mathbf{q}}^*(t)$ was calculated from that state. The resulting joint velocity was integrated using a control period of $\Delta t = 0.01$ s and

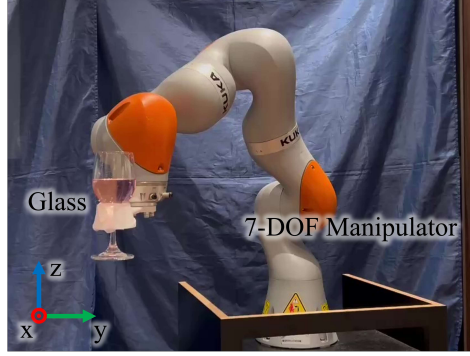


Fig. 7 Experimental setup using the KUKA LBR iiwa 7 R800, the transported object, and the coordinate frame.

Table 4 Initial conditions and goal positions for the transport experiments.

Scenario	$q(0)$ [rad]	x_s [m]	x_g [m]	$m(0)$
1	$[-0.524, 0.524, 0.000, -1.571, 0.000, 1.047, -0.524]^T$	$[0.473, -0.273, 0.360]^T$	$[0.473, 0.127, 0.360]^T$	0.147
2	$[-1.512, 0.300, 1.354, -1.459, -0.293, 1.615, -0.181]^T$	$[0.400, -0.190, 0.614]^T$	$[0.400, 0.210, 0.614]^T$	0.058

supplied to the robot as a joint-position command. Figure 7 shows the experimental setup.

Using the rotational component $J_\omega(t)$ of the six-dimensional Jacobian $J(t)$ defined in Section 2.1, the equality constraint for fixing the end-effector orientation is

$$J_\omega(t)\dot{q}(t) = \mathbf{0}. \quad (24)$$

Furthermore, to use joint motion that changes neither end-effector translation nor rotation for posture improvement, the null-space projection matrix in Section 3.3 was replaced in the real-robot evaluation by

$$N_6(t) = I - J(t)^+ J(t). \quad (25)$$

Apart from these changes, reference-position generation, the adaptive weight, and the joint constraints were identical to those in the proposed method.

Two experimental scenarios were considered: a standard scenario and a near-singularity scenario. The standard scenario comprised an initial configuration and goal position for which manipulability did not decrease substantially even when a straight-line reference trajectory was used. In the near-singularity scenario, the Jacobian loses rank and manipulability becomes zero at an intermediate configuration on the straight-line reference trajectory connecting the start and goal positions. Following this trajectory would therefore pass through a singular configuration during the motion. Table 4 lists the two experimental scenarios.

In addition to the proposed method, the comparison included a method that tracks a straight-line reference trajectory [10] and a method that permits deviation from the same reference trajectory [11]. The robot model, initial configuration, goal position, end-effector orientation constraint, joint-position and joint-velocity limits, and control

period were common to all three methods. Each method was executed once for each scenario. For the comparison methods, a straight-line trajectory connecting the start and goal positions was supplied as the reference trajectory. The principal parameters of the method in [10] were set to the values from the original paper: $\epsilon_{\text{NN}} = 0.01$, $c_0 = c_1 = 1$, and $c_2 = c_3 = 0.01$. For the method in [11], $\alpha = 5 \times 10^5$ and $Q_{\text{high}} = 10^5$ were used based on the original paper. A candidate range for Q_{low} was defined in view of the numerical scale of manipulability for the robot used in this study. From this range, the value yielding an actual path-length ratio closest to that of the proposed method was selected separately for each scenario, without using manipulability as a selection criterion. The selected values were 1650 for Scenario 1 and 1200 for Scenario 2, yielding actual path-length ratios of 1.08 and 1.06, respectively.

To evaluate the translational motion capability available during transport, translational manipulability under the orientation constraint, $m_c(t)$, was introduced in addition to the translational manipulability $m(t)$ defined in Section 2.1 [31]. The orthogonal projection matrix onto the joint-velocity space satisfying the fixed-orientation condition is

$$N_\omega(t) = I - J_\omega(t)^+ J_\omega(t), \quad (26)$$

and translational manipulability under the orientation constraint is defined as

$$m_c(t) = \sqrt{\det(J_v(t)N_\omega(t)J_v(t)^T)}. \quad (27)$$

$N_\omega(t)$ constrains rotational motion only and differs from $N_6(t)$, which preserves both translation and rotation. Both $m(t)$ and $m_c(t)$ were computed after the experiments from the same measured joint-position sequence. Motion generation used only $m(t)$ and its gradient, whereas $m_c(t)$ was used solely to evaluate translational motion capability under the fixed-orientation constraint. The orientation error was defined as the relative rotation angle from the initial end-effector orientation and evaluated in degrees. These time-series metrics were evaluated for both scenarios with the proposed method.

5.2 Experimental Results and Discussion

In this section, the standard and near-singularity scenarios described in Section 5.1 are denoted as Scenarios 1 and 2, respectively.

Figure 8 shows the real-robot motion generated by the proposed method in each scenario. In both scenarios, the end-effector deviated from the straight line connecting the start and goal positions before reaching the goal, while its z -axis direction was approximately maintained. These results confirm that the proposed method generated a transport path online according to the current state while maintaining the end-effector orientation.

Figure 9 shows the time histories of unconstrained translational manipulability $m(t)$ and orientation-constrained translational manipulability $m_c(t)$ in the two scenarios. Because the two indices have different definitions and numerical scales, their absolute magnitudes are not directly compared.

In Scenario 1, unconstrained translational manipulability $m(t)$ improved slightly during the first half of the motion from its initially high value and then decreased

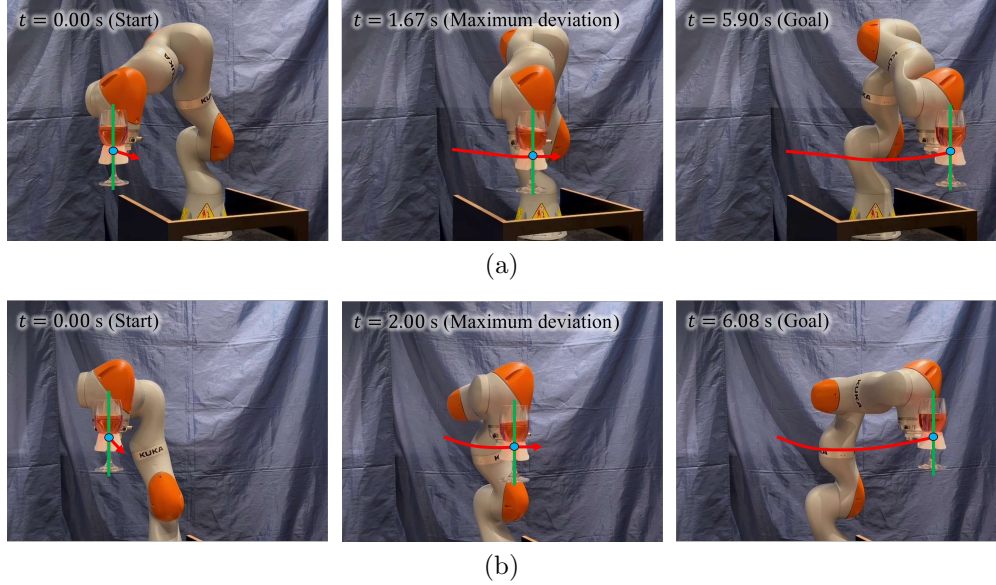


Fig. 8 Snapshots of transport motions generated by the proposed method with a fixed end-effector orientation in (a) Scenario 1 and (b) Scenario 2. From left to right, the columns show Start, Maximum deviation, and Goal, where Maximum deviation denotes the instant at which the distance from the end-effector position to the straight line connecting the start and goal positions was greatest. Blue markers indicate the current end-effector positions, green lines indicate the end-effector z -axis directions, red lines indicate the paths traversed, and red arrows indicate the directions of motion.

as the end-effector approached the goal. Under the specified goal position and fixed end-effector orientation, the remaining redundancy provided limited scope for further improvement in manipulability, which likely contributed to the decrease near the end of the motion. These results show that the manipulator reached the specified goal position while improving manipulability during the motion. In Scenario 2, manipulability improved without falling below its initial value, confirming that path modification and the use of redundant posture by the proposed method were effective for transport from a low-manipulability state.

Translational manipulability under the orientation constraint, $m_c(t)$, reached its maximum in the first half of the motion and then decreased in Scenario 1, whereas it increased from the initial state to the end of the motion in Scenario 2. Thus, on the trajectories generated from $m(t)$ in these experiments, $m_c(t)$ exhibited corresponding increasing and decreasing trends. This result does not establish a general relationship between the two indices and is presented only as an observation under the present experimental conditions.

Table 5 lists representative evaluation metrics for the three methods.

In Scenario 1, the proposed method outperformed both comparison methods in the mean, minimum, and maximum values of both unconstrained and orientation-constrained translational manipulability. In particular, for the method in [11], selecting $Q_{\text{low}} = 1650$ to make its actual path-length ratio as close as possible to that of the proposed method resulted in an identical ratio of 1.08 for both methods. The higher

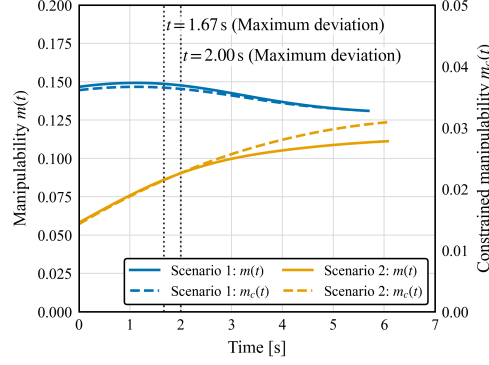


Fig. 9 Time histories of translational manipulability in each scenario with the proposed method. The solid lines show unconstrained translational manipulability $m(t)$ on the left vertical axis, and the dashed lines show translational manipulability under the orientation constraint $m_c(t)$ on the right vertical axis. The vertical dotted lines indicate the maximum-deviation instants shown in Figure 8.

Table 5 Experimental comparison of the proposed method with the methods in [10] and [11] under a fixed end-effector orientation.

Scenario	Method	Mean m	Minimum m	Maximum m	$L/L_{\min}$	Mean m_c	Minimum m_c	Maximum m_c
1	Proposed	0.142	0.131	0.149	1.08	0.0351	0.0327	0.0367
1	Jin et al. [10]	0.131	0.122	0.147	1.00	0.0327	0.0308	0.0361
1	Dufour and Suleiman [11]	0.139	0.129	0.148	1.08	0.0345	0.0324	0.0364
2	Proposed	0.094	0.058	0.111	1.07	0.0247	0.0143	0.0309
2	Jin et al. [10]	0.086	0.058	0.112	1.00	0.0230	0.0133	0.0313
2	Dufour and Suleiman [11]	0.107	0.058	0.117	1.06	0.0299	0.0143	0.0324

Note: Both comparison methods optimize manipulability. Jin et al. [10] impose trajectory tracking as an equality constraint, whereas Dufour and Suleiman [11] allow deviations from the reference trajectory.

manipulability achieved by the proposed method under this condition suggests that, in this scenario, generating a path from the current state without using a fixed reference trajectory produced an end-effector path effective for manipulability improvement. In Scenario 2, which began at low manipulability, the proposed method achieved higher mean values for both manipulability indices than the method in [10]; the minimum unconstrained translational manipulability was comparable, and the minimum orientation-constrained translational manipulability was higher. The maxima of both indices were slightly lower than those of that method. Compared with the method in [11], the proposed method had lower means and maxima for both indices and comparable minima. Although the proposed method does not maximize every manipulability metric, it reached the goal while gradually improving manipulability without falling below its initial value, as shown in Figure 9.

These results are consistent with the design of the proposed method, which balances manipulability improvement and goal attainment through an adaptive weight based on the current state and predicted potential for improvement, together with attenuation of the reference velocity near the goal. Under the present experimental conditions, the method generated transport paths that balanced manipulability improvement and goal attainment in two scenarios with different initial manipulability values.

The maximum orientation error relative to the initial end-effector orientation was 0.000652° in Scenario 1 and 0.0178° in Scenario 2, and the orientation change during transport was small in both scenarios. These results confirm that, under the present experimental conditions, the initial end-effector orientation was approximately maintained while the end-effector path was modified.

To evaluate real-time feasibility, the computation time per control cycle was recorded for the proposed method. The mean computation times were 0.458 ms in Scenario 1 and 0.448 ms in Scenario 2, and the maximum computation times were 1.09 and 1.16 ms, respectively. The computation time remained below 10 ms in every control cycle in both scenarios, confirming that the proposed method could be executed continuously at 100 Hz.

These results demonstrate that the proposed method generated transport paths online without a predefined fixed reference trajectory while satisfying the end-effector orientation and joint constraints. This evaluation was limited to the specified initial configurations and goal positions and does not guarantee goal attainment from an arbitrary initial state or global path optimality.

6 Conclusion

This study proposed an online motion generation method for point-to-point motion of a redundant manipulator that adjusts the balance between manipulability improvement and goal attainment according to the state. A reference ellipsoid was updated sequentially by subtracting the cumulative end-effector path length from a path-length budget, and a reference position that maximized the manipulability predicted by a local first-order approximation within this region was obtained analytically. An adaptive weight was also defined from the predicted improvement normalized by the current manipulability and the reference distance, thereby determining the priority of improvement without directly using the global maximum manipulability in the weight calculation. These components were integrated with reference velocities that decay near the goal and null-space posture improvement into a velocity-level QP subject to joint constraints, forming a framework that generates a path without a predefined end-effector reference trajectory.

In simulation, the allowable detour ratio was selected from the trade-off between manipulability improvement and actual path length in Scenario A. Comparison with a setting in which the weight in the manipulability direction was zero demonstrated that modifying the end-effector path, in addition to null-space posture improvement, was effective in suppressing decreases in manipulability during the motion. In comparison with fixed weights, the adaptive weight used a common set of design parameters across all scenarios, limited path modification under high-manipulability conditions, prioritized improvement under low-manipulability conditions, and reached the goal in every scenario. The results demonstrated reduced reliance on scenario-specific tuning of a fixed manipulability weight.

In the real-robot evaluation, the proposed method balanced manipulability improvement and goal attainment while approximately maintaining the end-effector orientation. In Scenario 1, it achieved higher manipulability than two comparison

methods, including one whose parameter was selected to make its actual path-length ratio as close as possible to that of the proposed method. In Scenario 2, which began at low manipulability, the comparison method configured using the same selection criterion achieved higher mean and maximum manipulability; nevertheless, the proposed method also reached the goal while improving manipulability without allowing it to fall below its initial value. On the experimental trajectories generated using $m(t)$, translational manipulability under the orientation constraint, $m_c(t)$, exhibited corresponding increasing and decreasing trends. This is an experimental observation from the two specified scenarios and does not establish a general relationship between the two indices. The computation time remained below 10 ms in every control cycle, confirming real-time feasibility at 100 Hz.

The path-length budget in the proposed method is a design quantity that defines the admissible region for the reference position and does not guarantee a strict upper bound on the actual path length. Moreover, the evaluation was limited to the specified scenarios and does not guarantee goal attainment from an arbitrary initial state or global optimality. Future work will examine applicability and goal reachability over a broader range of initial states, integration with global planning methods, and extension to environments containing obstacles.

Acknowledgements. This work was supported by JST SPRING, Grant Number JPMJSP2103.

Declarations

Author contributions. Ryuji Nakagawa designed the method, conducted the simulations and experiments, and wrote the manuscript. Ryosuke Tasaki supervised the study. All authors read and approved the final manuscript.

Funding. This work was supported by JST SPRING, Grant Number JPMJSP2103.

Availability of data and materials. No datasets were generated or analysed during the current study.

Competing interests. The authors declare no competing interests.

References

- [1] Yoshikawa, T.: Manipulability of Robotic Mechanisms. *The International Journal of Robotics Research* **4**(2), 3–9 (1985) <https://doi.org/10.1177/027836498500400201>
- [2] Chen, F., Selvaggio, M., Caldwell, D.G.: Dexterous Grasping by Manipulability Selection for Mobile Manipulator With Visual Guidance. *IEEE Transactions on Industrial Informatics* **15**(2), 1202–1210 (2019) <https://doi.org/10.1109/TII.2018.2879426>
- [3] Kennel-Maushart, F., Poranne, R., Coros, S.: Manipulability optimization for multi-arm teleoperation. In: 2021 IEEE International Conference on Robotics and

- Automation (ICRA), pp. 3956–3962 (2021). <https://doi.org/10.1109/ICRA48506.2021.9561105>
- [4] Selvamuthu, M.G., Abe, K., Tadakuma, K., Tadakuma, R.: Design and development of a compact small-scale spherical gear mechanism with manipulability-based singularity avoidance. *ROBOMECH Journal* **13**(1), 1 (2026) <https://doi.org/10.1186/s40648-025-00330-5>
 - [5] Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., Dillmann, R.: Manipulability analysis. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pp. 568–573 (2012). <https://doi.org/10.1109/HUMANOIDS.2012.6651576>
 - [6] Vahrenkamp, N., Asfour, T., Dillmann, R.: Robot placement based on reachability inversion. In: 2013 IEEE International Conference on Robotics and Automation, pp. 1970–1975. IEEE, Karlsruhe, Germany (2013). <https://doi.org/10.1109/ICRA.2013.6630839>
 - [7] Zhang, H., Sheng, Q., Hu, J., Sheng, X., Xiong, Z., Zhu, X.: Cooperative Transportation With Mobile Manipulator: A Capability Map-Based Framework for Physical Human–Robot Collaboration. *IEEE/ASME Transactions on Mechatronics* **27**(6), 4396–4405 (2022) <https://doi.org/10.1109/TMECH.2022.3155601>
 - [8] Yao, H., Laha, R., Figueredo, L.F.C., Haddadin, S.: Enhanced Dexterity Maps (EDM): A New Map for Manipulator Capability Analysis. *IEEE Robotics and Automation Letters* **9**(2), 1628–1635 (2024) <https://doi.org/10.1109/LRA.2023.3345695>
 - [9] Zhang, Y., Yan, X., Chen, D., Guo, D., Li, W.: QP-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators. *Nonlinear Dynamics* **85**(1), 245–261 (2016) <https://doi.org/10.1007/s11071-016-2681-9>
 - [10] Jin, L., Li, S., La, H.M., Luo, X.: Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. *IEEE Transactions on Industrial Electronics* **64**(6), 4710–4720 (2017) <https://doi.org/10.1109/TIE.2017.2674624>
 - [11] Dufour, K., Suleiman, W.: On Maximizing Manipulability Index while Solving a Kinematics Task. *Journal of Intelligent & Robotic Systems* **100**(1), 3–13 (2020) <https://doi.org/10.1007/s10846-020-01171-7>
 - [12] Yang, X., Zhao, Z., Ma, B., Xu, Z., Zhao, J., Liu, H.: Kinematic and Dynamic Manipulability Optimizations of Redundant Manipulators Based on RNN Model. *IEEE Transactions on Industrial Informatics* **20**(4), 5763–5773 (2024) <https://doi.org/10.1109/TII.2023.3334305>
 - [13] Xie, Z., Li, M., Sun, W., Cao, B., Liu, Y., Wang, Z., Ji, Y., Liu, H., Ma, B., Wu, Z.:

- Enhancing Safety and Manipulability of Redundant Manipulators: Accelerated Motion Generation in Dynamic Environments. *IEEE Robotics and Automation Letters* **10**(9), 8642–8649 (2025) <https://doi.org/10.1109/LRA.2025.3579633>
- [14] Haviland, J., Corke, P.: NEO: A Novel Expeditious Optimisation Algorithm for Reactive Motion Control of Manipulators. *IEEE Robotics and Automation Letters* **6**(2), 1043–1050 (2021) <https://doi.org/10.1109/LRA.2021.3056060>
 - [15] Marić, F., Limoyo, O., Petrović, L., Ablett, T., Petrović, I., Kelly, J.: Fast Manipulability Maximization Using Continuous-Time Trajectory Optimization. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8258–8264 (2019). <https://doi.org/10.1109/IROS40897.2019.8968441>
 - [16] Pardi, T., Ortenzi, V., Fairbairn, C., Pipe, T., Esfahani, A.M.G., Stolkin, R.: Planning Maximum-Manipulability Cutting Paths. *IEEE Robotics and Automation Letters* **5**(2), 1999–2006 (2020) <https://doi.org/10.1109/LRA.2020.2970949>
 - [17] Shen, H., Xie, W.-F., Tang, J., Zhou, T.: Adaptive Manipulability-Based Path Planning Strategy for Industrial Robot Manipulators. *IEEE/ASME Transactions on Mechatronics* **28**(3), 1742–1753 (2023) <https://doi.org/10.1109/TMECH.2022.3231467>
 - [18] Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research* **5**(1), 90–98 (1986) <https://doi.org/10.1177/027836498600500106>
 - [19] Kim, D.H., Shin, S.: Local path planning using a new artificial potential function composition and its analytical design guidelines. *Advanced Robotics* **20**(1), 115–135 (2006) <https://doi.org/10.1163/156855306775275530>
 - [20] Zhao, M., Lv, X.: Improved Manipulator Obstacle Avoidance Path Planning Based on Potential Field Method. *Journal of Robotics* **2020**, 1–12 (2020) <https://doi.org/10.1155/2020/1701943>
 - [21] Szczepanski, R., Tarczewski, T., Erwinski, K.: Energy Efficient Local Path Planning Algorithm Based on Predictive Artificial Potential Field. *IEEE Access* **10**, 39729–39742 (2022) <https://doi.org/10.1109/ACCESS.2022.3166632>
 - [22] Kovács, B., Szayer, G., Tajti, F., Burdelis, M., Korondi, P.: A novel potential field method for path planning of mobile robots by adapting animal motion attributes. *Robotics and Autonomous Systems* **82**, 24–34 (2016) <https://doi.org/10.1016/j.robot.2016.04.007>
 - [23] Zhu, T., Mao, J., Han, L., Zhang, C., Yang, J.: Real-Time Dynamic Obstacle Avoidance for Robot Manipulators Based on Cascaded Nonlinear MPC With Artificial Potential Field. *IEEE Transactions on Industrial Electronics* **71**(7), 7424–7434 (2024) <https://doi.org/10.1109/TIE.2023.3306405>

- [24] Becker, M., Caspers, P., Lilge, T., Haddadin, S., Müller, M.A.: Informed circular fields: A global reactive obstacle avoidance framework for robotic manipulators. *Frontiers in Robotics and AI* **11**, 1447351 (2025) <https://doi.org/10.3389/frobt.2024.1447351>
- [25] Nakagawa, R., Kimura, H., Tasaki, R.: Gradient-Guided Prediction of High Manipulability Configurations for Potential Field Surface-Following Path Planning. In: 2026 IEEE International Conference on Industrial Technology (ICIT), pp. 1–6 (2026). <https://doi.org/10.1109/ICIT64854.2026.11491583>
- [26] Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2997–3004 (2014). <https://doi.org/10.1109/IROS.2014.6942976>
- [27] Park, J., Chung, W., Youm, Y.: Computation of Gradient of Manipulability for Kinematically Redundant Manipulators Including Dual Manipulators System. *Transactions on Control, Automation and Systems Engineering* **1**(1), 8–15 (1999)
- [28] Haviland, J., Corke, P.: Manipulator Differential Kinematics: Part 2: Acceleration and Advanced Applications. *IEEE Robotics & Automation Magazine* **31**(4), 159–169 (2024) <https://doi.org/10.1109/MRA.2023.3270221>
- [29] Ma, B., Xie, Z., Zhan, B., Jiang, Z., Liu, Y., Liu, H.: Actual Shape-Based Obstacle Avoidance Synthesized by Velocity–Acceleration Minimization for Redundant Manipulators: An Optimization Perspective. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(10), 6460–6474 (2023) <https://doi.org/10.1109/TSMC.2023.3283266>
- [30] KUKA Robotics GmbH: KUKA LBR Iiwa Brochure (EN). [Online]. Available: <https://www.kuka.com>
- [31] Dimeas, F., Moulianitis, V.C., Papakonstantinou, C., Aspragathos, N.: Manipulator performance constraints in Cartesian admittance control for human-robot cooperation. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3049–3054 (2016). <https://doi.org/10.1109/ICRA.2016.7487469>