mathematics

Article

Toward Optimal Robot Machining Considering the Workpiece Surface Geometry in a Task-Oriented Approach

Aleš Hace

Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, SI-2000 Maribor, Slovenia; ales.hace@um.si; Tel.: +386-2-2207301

Abstract: Robot workpiece machining is interesting in industry as it offers some advantages, such as higher flexibility in comparison with the conventional approach based on CNC technology. However, in recent years, we have been facing a strong progressive shift to custom-based manufacturing and low-volume/high-mix production, which require a novel approach to automation via the employment of collaborative robotics. However, collaborative robots feature only limited motion capability to provide safety in cooperation with human workers. Thus, it is highly necessary to perform more detailed robot task planning to ensure its feasibility and optimal performance. In this paper, we deal with the problem of studying kinematic robot performance in the case of such manufacturing tasks, where the robot tool is constrained to follow the machining path embedded on the workpiece surface at a prescribed orientation. The presented approach is based on the well-known concept of manipulability, although the latter suffers from physical inconsistency due to mixing different units of linear and angular velocity in a general 6 DOF task case. Therefore, we introduce the workpiece surface constraint in the robot kinematic analysis, which enables an evaluation of its available velocity capability in a reduced dimension space. Such constrained robot kinematics transform the robot's task space to a two-dimensional surface tangent plane, and the manipulability analysis may be limited to the space of linear velocity only. Thus, the problem of physical inconsistency is avoided effectively. We show the theoretical derivation of the proposed method, which was verified by numerical experiments.

Keywords: robotics; automation; robot machining; workpiece surface polishing; collaborative robot; manipulability; complex surface geometry; motion planning

MSC: 70B15

check for updates

Citation: Hace, A. Toward Optimal Robot Machining Considering the Workpiece Surface Geometry in a Task-Oriented Approach. *Mathematics* **2024**, *12*, 257. https://doi.org/10.3390/math12020257

Academic Editor: Daniel-Ioan Curiac

Received: 27 November 2023
Revised: 10 January 2024
Accepted: 11 January 2024
Published: 12 January 2024

1. Introduction

Nowadays, robotics is an indispensable technology in many industries, especially in manufacturing, since it represents a major building block for fully automated production lines, such as in the automotive industry. Typically, standard applications of industrial robots are designed such that they perform well-defined repetitive tasks of manipulation, assembly, palletizing, welding, painting, etc. [1]. However, on the other hand, a shift from high-volume/low-mix to low-volume/high-mix and custom-based production impacts manufacturing seriously. It requires robots that are easier to install, program, and operate in order to increase robotization in small and medium-sized enterprises, where collaborative and more flexible automation appears to be a useful option. Thus, collaborative robots (cobots) not only provide safe physical coexistence, interaction, and cooperation in a common workspace with human workers [2,3] but are also easier to use. They are gaining popularity in industries with a growing demand for highly customized products since they increase manufacturing flexibility, filling the gap between fully automated systems and manual production [4,5]. However, in order to guarantee safe physical human–robot interaction, collaborative robots are designed with substantially reduced capabilities of speed and force/torque in comparison with standard industrial robot arms [6].

In addition to the above-mentioned industrial robot applications, there are also others, such as more complex robot machining [7] or milling [8], deburring [9], surface finishing [10], grinding [11], polishing [12,13], hammer-peening [14,15], etc. An excellent review of the state-of-the-art and the perspectives on the use of industrial robots in mechanical machining processes is presented in [16]. In some processes, the subtracting manufacturing technology requires a significant machining force in the tool feed direction for material removal, and in some, the machining force is required in the direction only toward the workpiece surface. The technological process of metal surface treatment with hammer-peening, which is applied in the tool- or mold- or die-making industry, is such a cold forging process, in which a ball made of a carbide solid is struck with a high frequency on the metal surface of the workpiece by a micro-forging hammer tool [17]. Hammer-peening improves the smoothness of the surface, hardens it, and eliminates internal stresses. In this manufacturing process, the machine hammer-peening tool (MHP tool) moves freely along the surface of the workpiece (that may have a complex geometry) following the machining path, which connects all surface points of the patch area to be forged. The working angle of the MHP tool to the workpiece is usually determined as perpendicular to its surface along the direction of travel of the tool. However, such a surface finishing process is complex and thus a very wasteful process, which has a large impact on the cost of the overall processing in the tool-making industry. It demands combining both a collaborative, intelligence-based, cooperative human–robot-based technological approach, as in the robotic polishing application [18]. Thus, it may involve not only robotized solutions, e.g., such as [19,20], but also collaborative automation with a proper flexibility due to the customized high mix/low volume nature of the production type in the tooling industry. Therefore, cobots can be considered as a justified replacement for traditional industrial robots in such applications.

However, due to low-power built-in cobot actuators (in comparison with standard industrial robot arms), their introduction in a machining process requires more careful trajectory planning in order to ensure the feasibility of the robot task, especially in the case of manufacturing processes with complex continuous paths where the complexity of robot path planning increases significantly [21]. Optimal relative workpiece/robot placement and robot path/trajectory planning considering this issue thus become even more important in order to provide a rapid setup of a robotic system in flexible high mix/low volume applications. Such planning problems have been interesting research topics for many years, and their application in practice still attracts attention in the research community. In general, path/trajectory planning is important for industrial robot applications since there is a strong interest in reducing the time interval of production cycles to provide optimal robot energy performance and task feasibility under the robot's joint physical limits [16,22]. The researchers optimized the location of the robot to generate maximum task-space velocity with minimum joint velocities [23]. For robot-to-workpiece placement for large-scale welding systems [24], the authors generated a kinematic performance map based on a kinetostatic condition index that was used to optimize robot configurations in a polishing application [25], introduced a custom index for robot-based placement optimization demonstrated in a trim application in shoe manufacturing [26], and optimized a workpiece placement for the robotic operation in challenging manufacturing tasks [27,28] and surface finishing [21,29]. An interesting new optimization approach was also introduced to maximize the available velocities of the end-effector during a task execution of path following in robot machining called the decomposed twist feasibility method [30].

If we consider the machining of workpieces with a complex surface, the difficulties in path and trajectory planning significantly increase. The machining efficiency and quality in the processing of so-called free-form surfaces significantly depend on tool paths in Cartesian space and the corresponding trajectory in joint space as well. An excellent review of the most interesting state-of-the-art solutions in this area has been given in [31]. Although path and feed-rate planning and trajectory planning are closely related and all coupled together, the complexity of the overall planning problem is such that most researchers approach them rather separately. It is also desired to keep the tool velocity as close to the maximum machining velocity as possible and simultaneously satisfy the robot's motion limits, which are, in the case of cobots, significantly lower in comparison with standard industrial robots. Additional consideration is related to independent control of the robot end-effector position and orientation. In free-form machining applications, the tool orientation is usually defined using the surface normal vector while following the position of the machining path. In order to provide optimal robot machining of curved free-form surfaces, it is thus necessary to consider workpiece surface geometry. The authors in [32] involve differential surface curve analysis for tool path and Cartesian trajectory planning. Surface curvature characteristics may also be considered for tool path generation [33,34] and region partitioning [35,36] of complex surfaces.

There have been several attempts at trajectory optimization for an industrial robot with different objectives. However, the main well-known basic concept that enables analysis of robot motion capability is called manipulability [37]. It is the most important and commonly used concept not only for robot mechanism design [38,39] but also for robot task-, path- and trajectory-planning in various robotics applications [40–49] since it can provide information about the distance to a singular robot configuration, motion capability of the robot, and optimal motion direction in a robot's operational task space. The pioneering work was presented by Yoshikawa in 1985 [50], who introduced a qualitative method for the assessment of robot motion capabilities based on the so-called manipulability ellipsoid, which was derived from a robot velocity kinematics description by the application of the singular value decomposition of the associated Jacobian matrix and Euclidean norm metrics. The introduced manipulability index was proportional to the ellipsoid volume, and it should measure how easy or difficult it is for a robot to move in its Cartesian operational workspace. It may also represent a distance to the singular configuration of the robotic arm. In addition to the volume of the ellipsoid, other indices also derived from the ellipsoid appeared later, such as minimal singular value and condition number [51]. However, the operational space twist vector involves different units of linear and angular velocity; in addition, robot joints can also be of different types. These complicate the manipulability analysis from the point of view of physical consistency [52,53]. Thus, in order to avoid the problem of dimensional dependence when both the position and orientation of the robot end-effector are included in kinematic equations, new manipulability indices were introduced. They were based on the introduction of auxiliary points on the end-effector that provided additional linear velocity information and a redundant formulation of the velocity equations instead of the combination of linear and angular velocities [54]. This approach is interesting since the resulting Jacobian is dimensionally homogeneous, and the point velocities can have some meaning to the designer [55]. However, in this point-based approach, the determination of the auxiliary points is arbitrary, and the measure of the obtained dimensionally homogeneous Jacobian matrix is not invariant with respect to changes in the auxiliary points used to express the end-effector velocity. Alternatively, some researchers proposed to normalize the Jacobian matrix in different fashions, such as pre-multiplication or post-multiplication with diagonal matrices that contain the desired maximum values for the end-effector twist components and the maximum available actuator speeds, respectively, or by using the so-called "natural length" or "characteristic length" of a robotic manipulator, which lack a sound physical interpretation [55]. A general approach to the problem of dimensional non-homogeneous matrices in the velocity kinematics description presents the introduction of weighted norms [56]. However, the selection of the corresponding weighting matrices to set translational components of the Jacobian matrix in relation to the rotational components [43] is again arbitrary, e.g., manual selection of the weights in a task-dependent measure [57], or their selection can be based on the computation of the minimal principal angle between the translational and rotational subspace in a task-oriented approach [58]. Nevertheless, it has been shown that all such manipulability indices suffer from the nonexistence of "natural" metrics and, therefore, from non-invariance in the sense of the choice of the selected artificial metric functions, which is arbitrarily employed in their definition [59,60]. Although the translational and rotational operational velocities can be separated by exploiting and extending the concept of manipulability in both weak and strong senses [61], the problem remains. Another possibility to overcome the problem of a non-homogeneous Jacobian matrix is based on the apparent power concept of the robot mechanism, which results in a homogeneous formulation of the problem, regardless of having mixed units in the velocity kinematics description [62]. However, the relation of the proposed measure to the relevant robot operational quantity, such as velocity, is unclear. An alternative to manipulability ellipsoids is presented by manipulability polytopes [30,63–68], which can provide more accurate information about the operational space motion capability since they consider infinity norm metrics instead of Euclidean norm metrics. However, the problem of a dimensional non-homogeneous Jacobian matrix due to the unit inconsistency in robotics still presents a challenge since most approaches demonstrated limitations in terms of their physical interpretations, though task-oriented homogeneous Jacobians and associated performance indices showed some promising results and potential for further development [30,69].

The main motivation for the research presented in this paper lies in the fact that the involvement of cobots in robot machining applications raises several issues, among them is also robot task feasibility concerning low cobot motion capability. For example, execution of the machining path can result in a trajectory with excessive joint speed, which is beyond the actuator speed limit. What is the optimal machining path and maximum machining speed? This challenge is especially emphasized in the case of freeform surface machining. Thus, the optimal machining task design, including path and trajectory planning, which will provide task feasibility, should be performed prior to robot task execution. In the previous paragraph, we discussed the manipulability concept that can be used for path/trajectory planning. However, all the manipulability-related performance indices above, and many others not mentioned, are based on the Jacobian matrix of the velocity kinematics description. In this paper, we focus on the problem of path following velocity performance in robot machining of a workpiece with complex geometry, similar to our previous work [30], where we considered a workpiece with a predefined path. Now, we propose a task-oriented robot kinematics description, which considers motion constraints imposed by the workpiece surface geometry explicitly; thus, a priori knowledge about the machining path is not required to evaluate the available velocity performance. The theoretical development results in the novel task-specific augmented inverse Jacobian matrix with incorporated motion constraints derived from the differential surface geometry of a workpiece. It represents the main contribution since the associated Jacobian matrix is homogeneous in units, and the task space is virtually reduced to the translational subspace. Furthermore, in comparison with [32–36], where workpiece surface geometry is involved in the path/trajectory planning in Cartesian space, we extend its consideration to the robot joint space via the manipulability concept to directly address the joints' velocity limits. We also show that the associated manipulability ellipsoid is reduced in dimension, such that we can perform velocity planning solely within the translational operational space projected on the two-dimensional surface tangent plane. Thus, the problem of the non-homogeneous twist space due to the mixed units is effectively avoided in a non-arbitrary way. The proposed task-oriented kinematics description can provide a basis not only for workpiece/robot placement optimization but also for optimal path/trajectory planning based on kinematic manipulability. We demonstrate the viability of the proposed approach by numerical experiments using two different workpiece examples.

The rest of the paper is organized as follows: in Section 2, we provide the necessary background information from the surface differential geometry; in Section 3, we theoretically develop the proposed approach; in Section 4, we show the numerical experiments; Section 5 presents a short discussion, and Section 6 concludes the paper.

Table 1 provides the standard nomenclature, which introduces the various symbols that appear in the equations presented in this paper.

Table 1. Standard nomenclature and abbreviations.

Notation	Meaning
$\boldsymbol{\mathcal{S}}$	Surface in a three-dimensional Euclidean space $\mathbb{R}^3$
u, v	Independent surface parameters
x, y, z	Cartesian position coordinates
P	Point on the surface
r	Position vector of a point on the surface
t	Independent time parameter
$\dot{x}, \dot{y}, \dot{z}$	Cartesian velocity coordinates
$\dot{r}$	Velocity vector of a curve on the surface
$(.)_x, (.)_y$	Partial derivatives of a vector w.r.t. x, y
s	Arc length of the curve
ds	Differential arc length of the curve
$\hat{n}$	Unit normal vector
I	First fundamental form matrix
E, F, G	Coefficients of the first fundamental form
II	Second fundamental form matrix
Π	Permuted second fundamental form matrix
L, M, N	Coefficients of the second fundamental form
W	Weingarten map matrix
K, H	Surface Gauss curvature, surface mean curvature
θ	Robot joint positions
$\dot{\theta}$	Robot joint velocities
$\dot{\theta}_C$	Robot joint velocities for constrained surface motion
J	Robot Jacobian matrix
J_T	Robot translational Jacobian matrix
J_R	Robot rotational Jacobian matrix
$\widetilde{J}_T$	Strong robot translational Jacobian matrix
$\widetilde{J}_R$	Strong robot rotational Jacobian matrix
P_T^n	Translational null-subspace projector into the robot joint space
P_R^n	Rotational null-subspace projector into the robot joint space
v	Robot end-effector linear velocity
ω	Robot end-effector angular velocity
ν	Robot end-effector twist
v_C	Robot end-eff. lin. vel. constrained on surface $\boldsymbol{\mathcal{S}}$
ω_C	Robot end-eff. ang. vel. constrained on surface $\boldsymbol{\mathcal{S}}$
S_C	Surface velocity constraint matrix
S_v	Surface lin. velocity constraint matrix
S_ω	Surface ang. velocity constraint matrix
$J_C^\#$	Augmented inverse robot Jacobian matrix with surface constraints
U, V	Orthonormal base of a matrix column space, row space (SVD result)
Σ	Matrix of singular values (SVD result)
SVD	Singular value decomposition
MHP	Machine hammer peening

2. Background

In this section, we provide some basic information about the differential geometry of surfaces and curves on surfaces that are relevant to this paper.

Let us consider surface $\boldsymbol{\mathcal{S}}$ in a three-dimensional Euclidean space that is parametrized by the position vector r

$$r = [x(u,v), y(u,v), z(u,v)], \tag{1}$$

as a function in the parametric form

$$r = r(u,v), \tag{2}$$

where u and v are independent parameters in a closed rectangle [70,71]. In this paper, we consider an explicit surface such that the z-coordinate can be expressed as a function of both the x- and y-coordinates:

$$z = f(x, y), \tag{3}$$

where f is at least twice the differentiable real-valued function, and the parametric form is derived by setting $x = u$ and $y = v$, with $[x, y] \in \mathcal{D}$, where $\mathcal{D}$ is a bounded connected domain in the real xy-plane. The surface is then given by

$$r = [x, y, f(x, y)]. \tag{4}$$

A curve on the surface is given by the parametrization $x = x(t)$, $y = y(t)$, and $z = f(x(t), y(t))$, where $t \in [0, T]$, and with $[x, y] \in \mathcal{D}$. Then $r = r(t)$ is a curve lying on and embedded in the surface (4). The tangent vector to the curve on the surface is evaluated by differentiating $r(t)$ with respect to the parameter t using the chain rule and is given by:

$$\dot{r}(t) = r_x \dot{x} + r_y \dot{y}, \tag{5}$$

where the subscripts x and y denote partial differentiation with respect to x and y such that $r_x = \partial r/\partial x$ and $r_y = \partial r/\partial y$, respectively, and the dot denotes differentiation with respect to the parameter t, such that $\dot{x} = dx/dt$ and $\dot{y} = dy/dt$. Note that (5) can be written in a form independent of the choice of parameter:

$$dr = r_x dx + r_y dy, \tag{6}$$

where $d(.)$ denotes a differential. The differential arc length of the curve ds is given as:

$$ds = \left|\frac{dr}{dt}\right| dt = |\dot{r}| dt = \sqrt{\dot{r} \circ \dot{r}} dt, \tag{7}$$

where $\circ$ stands for the vector dot product operator.

Let $P = P(x, y)$ be a point on the regular surface $\mathcal{S}$. Then, r_x and r_y are two independent surface tangent vectors at point P, which span a tangent plane. The tangent plane at point P on surface $\mathcal{S}$ can be considered as a union of all tangent vectors, which can be formed as a linear combination of r_x and r_y. The unit normal vector $\hat{n}$ on the surface at point P can be defined as

$$\hat{n} = \hat{n}(x, y) = \frac{r_x \times r_y}{\|r_x \times r_y\|}. \tag{8}$$

The unit normal vector is mapped from the tangent vectors r_x, r_y. It is perpendicular to the tangent plane, and, obviously, it is also orthogonal to both r_x and r_y. Thus, at any point on the surface we have a triple of vectors r_x, r_y, and $\hat{n}$, which are linearly independent on the regular surface $\mathcal{S}$. Note that the differential change in the unit normal vector can be written as:

$$d\hat{n} = \hat{n}_x dx + \hat{n}_y dy, \tag{9}$$

where $\hat{n}_x = \partial\hat{n}/\partial x$ and $\hat{n}_y = \partial\hat{n}/\partial y$ denote partial differentials of the unit normal vector with respect to x and y, respectively. The illustration of the tangent plane at a point on a curved surface along with the normal vector $\hat{n}$ and the other vectors r_x, r_y, n_x, n_y and $\dot{r}$ in the tangent plane are depicted by Figure 1.

The *first fundamental form* describes the method of measuring the distances on a surface, i.e., a surface metric. It determines the arc length of a curve on the surface and is defined as [71]:

$$\mathrm{I} : ds^2 = dr \circ dr. \tag{10}$$

If we consider (6), then it can be derived as follows

$$\mathrm{I}(dx, dy) : ds^2 = Edx^2 + 2Fdxdy + Gdy^2, \tag{11}$$

where the coefficients $E = r_x \circ r_x$, $F = r_x \circ r_y$, and $G = r_y \circ r_y$ are called the coefficients of the first fundamental form. Due to its quadratic-bilinear form of the coordinates' differentials on the surface, it is often presented by the symmetric matrix:

$$\mathrm{I} = \begin{bmatrix} E & F \\ F & G \end{bmatrix}, \tag{12}$$

which is a positive definite, i.e., $EG - F^2 > 0$ at regular points on the surface.

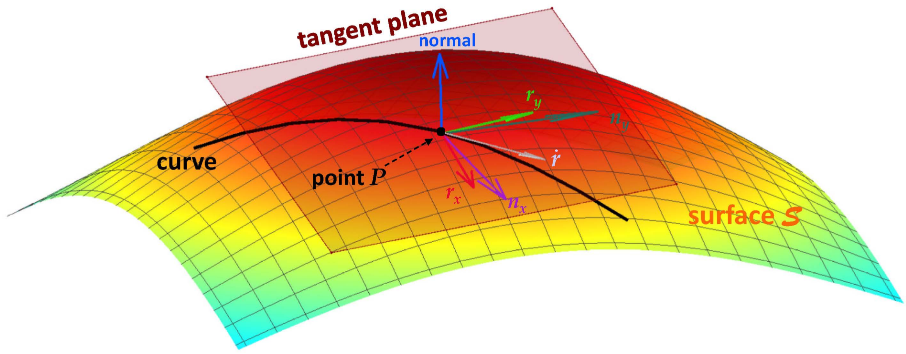


Figure 1. The illustration of the tangent plane at a point on a curved surface along with the normal vector and the other vectors in the tangent plane.

The *second fundamental form* characterizes the local structure of the surface shape in a neighborhood of a point. It describes how the surface deviates from the tangent plane. It can be defined as [71]

$$\mathrm{II} : (-dr \circ d\hat{n}), \tag{13}$$

It can be derived as

$$\mathrm{II}(dx, dy) : Ldx^2 + 2Mdxdy + Ndy^2, \tag{14}$$

where the coefficients $L = -r_x \circ \hat{n}_x = r_{xx} \circ \hat{n}$, $M = -r_x \circ \hat{n}_y = -r_y \circ \hat{n}_x = r_{xy} \circ \hat{n} = r_{yx} \circ \hat{n}$, and $N = -r_y \circ \hat{n}_y = r_{yy} \circ \hat{n}$ are called the coefficients of the second fundamental form. The matrix of the second fundamental form can be read as:

$$\mathrm{II} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}. \tag{15}$$

The differential $d\hat{n}$ is called a *Weingarten map*, which describes the change in the normal direction as we move from one point to another. It can be expressed in terms of the first derivatives of the position vector, and the partial derivatives of the unit normal vector can then be expressed in terms of the basis $\{r_x, r_y\}$ [71,72]:

$$\begin{bmatrix} \hat{n}_x & \hat{n}_y \end{bmatrix} = -\begin{bmatrix} r_x & r_y \end{bmatrix} W, \tag{16}$$

where W is the matrix, which can be calculated by (12) and (15) as:

$$W = \mathrm{I}^{-1} \cdot \mathrm{II} = \frac{1}{EG - F^2} \begin{bmatrix} GL - FM & GM - FN \\ EM - FL & EN - FM \end{bmatrix}, \tag{17}$$

where the operator $\cdot$ stands for matrix multiplication. The result is known as *Weingarten equations*:

$$\hat{n}_x = \frac{FM - GL}{EG - F^2} r_x + \frac{FL - EM}{EG - F^2} r_y, \tag{18}$$

$$\hat{n}_y = \frac{FN - GM}{EG - F^2} r_x + \frac{FM - EN}{EG - F^2} r_y. \tag{19}$$

which map from the tangent plane to the tangent plane, i.e., the vectors $\hat{n}_x$ and $\hat{n}_y$ are expressed as a linear combination of the tangent vectors r_x, r_y, respectively. Thus, they lie in the tangent plane and are orthogonal to the unit normal vector $\hat{n}$.

An important surface characteristic is also its curvature, which shows us how much the surface bends or deviates from a flat surface. The curvature measure should involve the rate at which surface $\boldsymbol{S}$ leaves the tangent plane to $\boldsymbol{S}$ at point P. A crucial tool to measure the curvature on a surface is the Weingarten map, which contains complete information about a surface's curvature. It can be used to derive scalar measures to show how 'curved' the surface is at some point. There are several measures for the curvature of a surface in $\mathbb{R}^3$. Gauss curvature K and mean curvature H on surface $\boldsymbol{S}$ at point P can be formulated by (20) and (21), respectively.

$$K = \det(W) = \frac{LN - M^2}{EG - F^2} \tag{20}$$

$$H = \frac{1}{2}\mathrm{trace}(W) = \frac{EN + GL - 2FM}{2(EG - F^2)} \tag{21}$$

Other measures, such as normal curvature and principal curvatures, are also possible [72].

3. Methodology

3.1. Problem Formulation

The serial robot arm manipulator can be described as an open chain mechanism connecting links and joints having n-degrees-of-freedom (DOF). The base link is normally considered fixed to the ground, and the final link, with a mounted end-effector, may operate freely in the robot task space. To provide all possible positions and orientations within its 3D workspace, the manipulator must have proper mechanism geometry with at least six DOFs. The velocity mapping from the joint space to the operational space can be described as the forward velocity kinematics model [37]:

$$\nu = J(\theta)\dot{\theta}, \tag{22}$$

where $J \in \mathbb{R}^{6xn}$ is the geometric Jacobian matrix in the robot's base frame, $\theta \in \mathbb{R}^n$ and $\dot{\theta} \in \mathbb{R}^n$ are the vectors of joint positions, which define the current mechanism configuration and their velocities, respectively. Furthermore, we denote the so-called twist vector with $\nu \in \mathbb{R}^6$

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{23}$$

which represents the combination of two physically different components: $v \in \mathbb{R}^3$ is the vector of the end-effector linear velocity, and $\omega \in \mathbb{R}^3$ is the vector of the end-effector angular velocity. The inverse velocity kinematics problem can be defined as finding a proper solution of the joint velocity $\dot{\theta}$ for the given twist ν. Note that the twist vector is not homogeneous in physical units.

In the following, we consider a nonredundant manipulator with $n = 6$ in a non-singular configuration. Although in general the joints can be of different types, we assume rotational joints without a significant loss of generality since most modern robot manipulators use only rotary joints with one DOF.

The most widely used kinematic measure that refers to manipulator manipulability and indicates how dexterous a robot is at a given configuration θ is derived from the condition [50]:

$$\dot{\theta}^T\dot{\theta} \leq 1, \tag{24}$$

which maps the unit sphere in the joint velocity space to the velocity manipulability ellipsoid in the operational task space:

$$\nu^T \left(JJ^T\right)^{-1} \nu \leq 1. \tag{25}$$

Here, we consider that the solution to the inverse velocity kinematics can be expressed by the full rank Jacobian as:

$$\dot{\theta} = J^{-1}\nu. \tag{26}$$

We can factorize the Jacobian matrix by singular value decomposition (SVD) into

$$J = U\Sigma V^T, \tag{27}$$

where Σ is a diagonal matrix with positive singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_6 > 0$ on the diagonal, and U and V are unitary matrices representing the orthonormal base of the Jacobian column space $Col(J)$ and the orthonormal base of the Jacobian row space $Row(J)$, respectively [73,74]. If we insert (27) into (25), it yields:

$$(U^T\nu)^T\Sigma^{-2}(U^T\nu) \leq 1, \tag{28}$$

where Σ^{-2} is a diagonal matrix with $\sigma_1^{-2}, \sigma_2^{-2}, \ldots, \sigma_6^{-2}$ on the diagonal. One can introduce a new variable $w = U^T\nu$, $w = [w_1, w_2, \ldots, w_6]^T$; then, (28) can be rewritten as:

$$w^T\Sigma^{-2}w = \sum_{i=1}^{6}\left(\frac{w_i}{\sigma_i}\right)^2 \leq 1, \tag{29}$$

which describes an ellipsoid geometrically with the axes' directions defined by the columns of the unitary matrix U, whereas the singular values σ_i determine the axes' lengths. The manipulability ellipsoid shows how far away the manipulator is from a singular configuration in a certain direction in the task space: if the manipulator is near the singular configuration, then the dexterity is low, and contrarily, if the manipulator is far from the singular configuration, then the dexterity is better. Using the manipulability ellipsoid, one can quantify how close a given robot posture is to a singularity. Note that the ellipsoid is described in the six-dimensional Euclidian space, which is not suitable for visual geometrical representation and is not intuitive.

Although the manipulability ellipsoid is quite popular among the research community since it enables the derivation of numerous manipulability indices [75], it suffers from serious limitations that arise due to the physical non-homogeneity of the twist vector and physical inconsistency of the Jacobian matrix [52,61].

3.2. The End-Effector Motion Constraints

During the workpiece machining, the tool tip must follow a certain path embedded in the workpiece surface. The tool path is constrained by position and orientation. In our case, we assumed that the tool must maintain orthogonal orientation to the curved workpiece surface, i.e., the tool orientation must be adjusted continuously while traveling along the path. In this section, we derive velocity synchronization, which arises due to the constraints described above.

The tool tip must follow the position of the path embedded on the workpiece surface. Let a particle refer to the tool tip. If a particle is moving in time along the path constrained on the workpiece surface, then the particle velocity can be expressed as:

$$v = \dot{r}, \tag{30}$$

If we consider (5), then the tool tip velocity can be derived as

$$v = \begin{bmatrix} r_x & r_y \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \tag{31}$$

Note that the tool linear velocity vector v is a linear combination of the tangent vectors r_x and r_y. Thus, it lies in a tangent plane of the workpiece surface. On the other side, (31) it can be read in a way that v is to be synchronized with the velocity vector in the xy-plane $\begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^T$ by the matrix S_v:

$$S_v = \begin{bmatrix} r_x & r_y \end{bmatrix}, \tag{32}$$

such that we can express:

$$v = S_v \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{33}$$

where $S_v \in \mathbb{R}^{3\times 2}$ is with full rank, i.e., $\mathrm{rank}(S_v) = 2$.

The tool orientation must be aligned by the normal vector perpendicular to the curved workpiece surface. Let a particle refer to the tool tip. If a particle is traveling along the path constrained on the workpiece surface, then, in each point of the path, the normal vector associated to the particle position can have a different orientation. In this case, the particle is not only translating but the associated surface normal vector can also rotate synchronously. We can observe the velocity of the unit normal vector $\hat{n} = \hat{n}(t)$, which can be described as:

$$\dot{\hat{n}} = \omega \times \hat{n}, \tag{34}$$

where $\times$ refers to the vector cross product operator. If we assume reasonably that the normal vector rotation about itself is zero, then the rotation axis is orthogonal to the unit normal vector, and it lies in the surface tangent plane of the observed point on the path. Then, the angular velocity of the unit normal vector can be derived as:

$$\omega = \hat{n} \times \dot{\hat{n}}. \tag{35}$$

Since the surface normal vector dependence can be expressed as $\hat{n} = \hat{n}(x, y)$, and by the application of the chain rule, we can derive:

$$\dot{\hat{n}} = \hat{n}_x \dot{x} + \hat{n}_y \dot{y}, \tag{36}$$

and furthermore

$$\omega = \begin{bmatrix} \hat{n} \times \hat{n}_x & \hat{n} \times \hat{n}_y \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{37}$$

where $\dot{x}$ and $\dot{y}$ are linear velocities in the x- and y-Cartesian coordinates, respectively. Now we can recall the Weingarten Equations (16)–(19) and after some manipulation one can derive:

$$\omega = \frac{\begin{bmatrix} r_x & r_y \end{bmatrix}}{\sqrt{EG - F^2}} \begin{bmatrix} M & N \\ -L & -M \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{\begin{bmatrix} r_x & r_y \end{bmatrix}}{\sqrt{EG - F^2}} (A \cdot \mathrm{II}) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{38}$$

where II is the second fundamental form matrix (15), and A is defined as:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{39}$$

The angular velocity vector ω is expressed by (38) as a linear combination of the tangent vectors r_x and r_y. Therefore, it belongs to the tangent plane of the workpiece surface, and it will not produce tool rotation about itself. Furthermore, the tool angular velocity vector ω will be synchronized with the linear velocity in the xy-plane $\begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^T$ by the matrix S_ω:

$$S_\omega = \frac{\begin{bmatrix} r_x & r_y \end{bmatrix}}{\sqrt{EG - F^2}} \Pi = \frac{S_v \Pi}{\sqrt{EG - F^2}}, \tag{40}$$

where we denote $\Pi = A \cdot \mathrm{II}$. Thus, we can express:

$$\omega = S_\omega \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{41}$$

where $S_\omega \in \mathbb{R}^{3\times 2}$, and its rank is $\text{rank}(S_\omega) \leq 2$.

In the following, we rewrite the velocity constraints described by Equations (33) and (41) such that they read as:

$$v_C = S_v \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{42}$$

and

$$\omega_C = S_\omega \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \tag{43}$$

where v_C and ω_C represent linear velocity and angular velocity constrained to the surface, respectively. Note that v_C (ω_C) belongs to the range-space of matrix S_v (S_ω). Furthermore, from (42), we can express

$$\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} = S_v^\dagger v_C, \tag{44}$$

where we consider $v_C = \left[\dot{x}_C, \dot{y}_C, \dot{z}_C\right]^T$, and the operator $(.)^\dagger$ stands for the Moore–Penrose pseudoinverse of a matrix, such that $S_v^\dagger = \left(S_v^T S_v\right)^{-1} S_v^T$ is the left pseudo-inverse of S_v. If we insert the last result into (43), it yields:

$$\omega_C = S_\omega S_v^\dagger v_C = S_C v_C. \tag{45}$$

Here, S_C represents the so-called *surface velocity constrained matrix*, which can be expressed as:

$$S_C = S_\omega S_v^\dagger = \frac{S_v \Pi S_v^\dagger}{\sqrt{EG - F^2}}. \tag{46}$$

Note that $S_C \in \mathbb{R}^{3\times 3}$, and its rank is $\text{rank}(S_C) \leq 2$. It maps the linear velocity vector from the surface tangent plane to the angular velocity vector on the surface tangent plane.

3.3. The Constrained Robot Kinematics Model

The velocity kinematics play a central role in robot manipulability analysis. In this section, we formulate the problem as seeking the inverse velocity kinematics solution of (22) and (23) considering the velocity constraints given by synchronization expressions for linear and angular velocity (42)–(46), which constrain the motion of the end-effector on the path embedded on the workpiece surface, such that the end-effector maintains normal orientation. If, furthermore, we consider the joint velocity limitation given by (24), then it will yield the constrained manipulability ellipsoid of a reduced dimension, as we will show later.

Similarly, as in [30], the Jacobian matrix is partitioned into two submatrices, i.e., the translational submatrix $J_T \in \mathbb{R}^{3\times n}$ and the rotational submatrix $J_R \in \mathbb{R}^{3\times n}$:

$$J = \begin{bmatrix} J_T \\ J_R \end{bmatrix}, \tag{47}$$

such that $J_T\dot{\theta} = v$ and $J_R\dot{\theta} = \omega$, where v and ω are the linear velocity vector and the angular velocity vector in the robot operational space, respectively. In the following, we derive a solution of the velocity inverse kinematics considering the partitioned matrix (47). Baksalaray [76,77] discussed the problem of the Moore–Penrose inverse of a partitioned matrix. The application of such a solution formula in the case of the Jacobian (47) can be written by the augmented Jacobian submatrices that incorporate additional restrictions [30,61,78]:

$$\widetilde{J}_T = J_T P_R^n. \tag{48}$$

$$\widetilde{J}_R = J_R P_T^n. \tag{49}$$

where $P_T^n = (I - J_T^\dagger J_T)$ and $P_R^n = (I - J_R^\dagger J_R)$ are orthogonal projection matrices into the translational and rotational null space, respectively. Here, I is the identity matrix of a proper dimension. Note that the definition of $\widetilde{J}_T$ ($\widetilde{J}_R$) imposes such a constraint that the angular (linear) velocity is additionally zero, and thus, it can be called a *strong* translational (rotational) Jacobian matrix [61]. The inverse of the partitioned squared Jacobian (n = 6, $J \in \mathbb{R}^{6\times 6}$) can now be expressed as the block matrix:

$$J^{-1} = \begin{bmatrix} J_T \\ J_R \end{bmatrix}^{-1} = [\widetilde{J}_T^\dagger \quad \widetilde{J}_R^\dagger], \tag{50}$$

where $\widetilde{J}_T^\dagger \in \mathbb{R}^{n\times 3}$ and $\widetilde{J}_R^\dagger \in \mathbb{R}^{n\times 3}$. We can apply the inverse Jacobian in the block form (50) in the computation of the inverse velocity kinematics solution as

$$\dot{\theta} = J^{-1}\begin{bmatrix} v \\ \omega \end{bmatrix} = [\widetilde{J}_T^\dagger \quad \widetilde{J}_R^\dagger]\begin{bmatrix} v \\ \omega \end{bmatrix} = \widetilde{J}_T^\dagger v + \widetilde{J}_R^\dagger \omega. \tag{51}$$

The solution consists of two joint velocity components: the first component $\widetilde{J}_T^\dagger v$ is responsible for the end-effector translational motion without affecting its rotation since it belongs to the null-space of J_R, whereas the second component $\widetilde{J}_R^\dagger v$ is responsible solely for the end-effector rotational motion with zero contribution to its translation since it belongs to the null-space of J_T. Indeed, if we apply the solution (51) in the forward velocity kinematics mapping (22), it yields the desired result:

$$\begin{aligned} J\dot{\theta} &= \begin{bmatrix} J_T \\ J_R \end{bmatrix}(\widetilde{J}_T^\dagger v + \widetilde{J}_R^\dagger \omega) = \begin{bmatrix} J_T((J_T P_R^n)^\dagger v + (J_R P_T^n)^\dagger \omega) \\ J_R((J_T P_R^n)^\dagger v + (J_R P_T^n)^\dagger \omega) \end{bmatrix} = \begin{bmatrix} J_T(P_R^n (J_T P_R^n)^\dagger v + P_T (J_R P_T^n)^\dagger \omega) \\ J_R(P_R^n (J_T P_R^n)^\dagger v + P_T (J_R P_T^n)^\dagger \omega) \end{bmatrix} = \\ &= \begin{bmatrix} J_T P_R^n (J_T P_R^n)^\dagger v + J_T P_T^n (J_R P_T^n)^\dagger \omega \\ J_R P_R^n (J_T P_R^n)^\dagger v + J_R P_T^n (J_R P_T^n)^\dagger \omega \end{bmatrix} = \begin{bmatrix} Iv + 0\omega \\ 0v + I\omega \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix} \end{aligned}. \tag{52}$$

Here, we exploit the well-known property of the Moore–Penrose pseudo inverse with the orthogonal projector, i.e., $(J_{T/R} P_{R/T}^n)^\dagger = P_{R/T}^n (J_{T/R} P_{R/T}^n)^\dagger$ [74,79]; consequently, $J_{T/R}\widetilde{J}_{T/R}^\dagger = I$ and $J_{T/R}\widetilde{J}_{R/T}^\dagger = 0$.

In the following, we replace v and ω in (51) with v_C and ω_C, respectively, and consider the expressions (42) and (43). It yields:

$$\dot{\theta} = \widetilde{J}_T^\dagger v_C + \widetilde{J}_R^\dagger \omega_C = (\widetilde{J}_T^\dagger S_v + \widetilde{J}_R^\dagger S_\omega)\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \tag{53}$$

Furthermore, we consider (44), and then (53) it can be rewritten as:

$$\dot{\theta}_C = (\widetilde{J}_T^\dagger S_v S_v^\dagger + \widetilde{J}_R^\dagger S_\omega S_v^\dagger) v_C = J_C^\# v_C, \tag{54}$$

where $\dot{\theta}_C$ stands for such joint velocities, which provide robot tool motion constrained on the workpiece surface, and $J_C^\# \in \mathbb{R}^{n\times 3}$ (with $n = 6$) denotes the so-called augmented inverse Jacobian matrix of the constrained kinematics,

$$J_C^\# = \widetilde{J}_T^\dagger S_v S_v^\dagger + \widetilde{J}_R^\dagger S_\omega S_v^\dagger, \tag{55}$$

which is homogeneous in units, i.e., all the elements of $J_C^\#$ are given in units of $(1/m)$, since $\widetilde{J}_T^\dagger$ is given in units of $(1/m)$, and S_v is a dimensionless matrix in the sense of units, whereas $\widetilde{J}_R^\dagger$ is a dimensionless matrix in the sense of units and S_ω is given in units of $(1/m)$. Note also that $S_v S_v^\dagger$ represents the orthogonal projector of the range space of matrix S_v (with the following properties $(S_v S_v^\dagger)^2 = (S_v S_v^\dagger)^T = S_v S_v^\dagger$), such that $v_C = S_v S_v^\dagger v$, and from (45), it follows directly that $S_\omega S_v^\dagger$ maps any v_C to the range space of matrix S_ω. Furthermore, if we consider the orthogonal decomposition of the operational space velocity vector v, such that:

$$v = v_C + v_C^{\perp}, \tag{56}$$

where $v_C^{\perp} = (I - S_v S_v^{\dagger})v$ is orthogonal to v_C, i.e., $v_C^{\perp} \perp v_C$, then it is easy to show that (54) can also be written as:

$$\dot{\theta}_C = (\widetilde{J}_T^{\dagger} S_v S_v^{\dagger} + \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger})v = J_C^{\#} v \tag{57}$$

since matrix $J_C^{\#}$ maps the velocity component vector $v_C^{\perp}$ to zero:

$$\begin{array}{c} J_C^{\#} v_C^{\perp} = (\widetilde{J}_T^{\dagger} S_v S_v^{\dagger} + \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger})(I - S_v S_v^{\dagger})v = (\widetilde{J}_T^{\dagger} S_v S_v^{\dagger}(I - S_v S_v^{\dagger}) + \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger}(I - S_v S_v^{\dagger}))v = \\ (\widetilde{J}_T^{\dagger}(S_v S_v^{\dagger} - (S_v S_v^{\dagger})^2) + \widetilde{J}_R^{\dagger} S_\omega (S_v^{\dagger} - S_v^{\dagger} S_v S_v^{\dagger}))v = (\widetilde{J}_T^{\dagger}(S_v S_v^{\dagger} - S_v S_v^{\dagger}) + \widetilde{J}_R^{\dagger} S_\omega (S_v^{\dagger} - S_v^{\dagger}))v = 0 \end{array}. \tag{58}$$

Finally, we show that the derived solution above provides such velocities in the operational robot space, which are constrained to the workpiece tangent plane. It is rather straightforward to derive that the forward velocity kinematics mapping (22) on $\dot{\theta}_C$ results as

$$J\dot{\theta}_C = J(J_C^{\#} v_C) = \begin{bmatrix} J_T \\ J_R \end{bmatrix} (\widetilde{J}_T^{\dagger} S_v S_v^{\dagger} + \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger}) v_C = \begin{bmatrix} J_T \widetilde{J}_T^{\dagger} S_v S_v^{\dagger} + J_T \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger} \\ J_R \widetilde{J}_T^{\dagger} S_v S_v^{\dagger} + J_R \widetilde{J}_R^{\dagger} S_\omega S_v^{\dagger} \end{bmatrix} v_C = \begin{bmatrix} S_v S_v^{\dagger} \\ S_\omega S_v^{\dagger} \end{bmatrix} v_C = \begin{bmatrix} v_C \\ \omega_C \end{bmatrix} \tag{59}$$

which confirms the solution described by (54), (55), and (57) provides the desired constrained motion (42) and (43) in the robot operational space. Furthermore, if we consider that the augmented inverse Jacobian of the constrained kinematics can also be represented as $J_C^{\#} = (J^{-1}S)S_v^{\dagger}$, where $S^T = \begin{bmatrix} S_v^T & S_\omega^T \end{bmatrix}$, it can also be shown easily that its rank equals $\mathrm{rank}(J_C^{\#}) = 2$; on the basis of the Sylvester rank inequality formula [80], it is possible to derive $\mathrm{rank}(J^{-1}S) = 2$ and in the following also $\mathrm{rank}((J^{-1}S)S_v^{\dagger}) = 2$, since the matrix $S_v^{\dagger}$ is of full rank, i.e., $\mathrm{rank}(S_v^{\dagger}) = 2$.

3.4. The Manipulability Ellipsoid of the Constrained Motion

Now, we can construct the manipulability ellipsoid of the constrained motion following a procedure that is similar to the steps determined by (24)–(29). Firstly, we consider the constraint in the joint space given by the condition (24), which is derived as:

$$\dot{\theta}^T \dot{\theta} = v^T (J_C^{\# T} J_C^{\#}) v \leq 1, \tag{60}$$

where $J_C^{\#}$ is defined by (55). In the next step, we can factorize matrix $J_C^{\#}$ by SVD into

$$J_C^{\#} = U \Sigma V^T. \tag{61}$$

However, due to the fact that the matrix is tall and its rank is deficient $r = \mathrm{rank}(J_C^{\#}) = 2$, the factorized matrices U, Σ, and V have the following structure:

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix},\ \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},\ \text{and } V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \tag{62}$$

respectively. Here, the columns of U_1 represent an orthonormal base of the column space $Col(J_C^{\#})$, whereas the columns of U_2 represent an orthonormal base of the left null space $Null(J_C^{\# T})$. Furthermore, the columns of V_1 represent an orthonormal base of the row space $Row(J_C^{\#})$, whereas the columns of V_2 represent an orthonormal base of the null space $Null(J_C^{\#})$. The (sub)matrices' dimensions are $\dim(U_1) = n \times r$, $\dim(U_2) = n \times (n - r)$, $\dim(V_1) = 3 \times r$, $\dim(V_2) = 3 \times (3 - r)$ and $\dim(\Sigma_1) = r \times r$, respectively. Note that U and V are unitary matrices with orthonormal columns, and Σ_1 is a diagonal square matrix with the positive singular values $\sigma_1 \geq \ldots \geq \sigma_r > 0$ on the diagonal. We can derive:

$$J_C^{\#} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T, \tag{63}$$

and further

$$J_C^{\#T} J_C^{\#} = (U_1 \Sigma_1 V_1^T)^T (U_1 \Sigma_1 V_1^T) = (V_1 \Sigma_1 U_1^T)(U_1 \Sigma_1 V_1^T) = V_1 \Sigma_1 U_1^T U_1 \Sigma_1 V_1^T = V_1 \Sigma_1^2 V_1^T. \tag{64}$$

Thus, (60) can be rewritten as

$$v^T V_1 \Sigma_1^2 V_1^T v \leq 1. \tag{65}$$

If we introduce a new variable of reduced order $w = V_1^T v$, then it yields the ellipsoid,

$$w^T \Sigma_1^2 w \leq 1 \tag{66}$$

which can be rewritten as

$$\frac{w_1^2}{\sigma_1^{-2}} + \frac{w_2^2}{\sigma_2^{-2}} \leq 1, \tag{67}$$

since $r = 2$, and thus $w^T = \begin{bmatrix} w_1 & w_2 \end{bmatrix}$, $\mathrm{diag}(\Sigma_1) = \begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix}$. Obviously, the constrained manipulability ellipsoid is reduced to an ellipse in the tangent plane of the workpiece surface.

4. Numerical Experiments

In this section, we show the results of some numerical experiments that we performed in order to verify the proposed approach of the robot manipulability evaluation in the case of the robot tool motion constrained by the workpiece surface. In the numerical experiments, we applied a collaborative robot arm and two workpiece models with different surface geometry. In the following, we firstly introduce the experimental setup. Then, we show the ellipsoids related to the velocity synchronization on the workpiece surface, which can be derived from (45)–(46). Next, we present the manipulability ellipsoids of the robot motion with the tool constrained to the surface geometry derived from (54)–(55), and it is described further in Section 3.4. Finally, we verify the drawn manipulability ellipsoids by the calculation of the robot tool velocity in selected directions. The numerical experiments were performed in the MATLAB computing environment [81].

The computation procedure applied in the numerical experiments can be summed up in the following main steps, which are divided into two phases:

- Phase A—workpiece surface analysis:
  1. Step: load the workpiece surface model with position data.
  2. Step: compute the surface partial derivatives of first order r_x, r_y.
  3. Step: compute the surface partial derivatives of second order r_{xx}, $r_{xy} = r_{yx}$, r_{yy}.
  4. Step: compute the surface normals by (8).
  5. Step: compute the coefficients of the first fundamental form E, F, G (see (11)).
  6. Step: compute the coefficients of the second fundamental form L, M, N (see (14)).
  7. Step (optionally): compute the surface curvatures by (20) and (21).
  8. Step: compute matrix S_v by (32) and then compute its pseudo inverse $S_v^{\dagger}$.
  9. Step: compute matrix S_ω by (40).
  10. Step: compute surface velocity constraint matrix S_C by (46).
- Phase B—task-oriented constrained robot kinematics analysis:
  11. Step: set up the robot and the workpiece position.
  12. Step: compute the robot inverse kinematics solutions for the given workpiece surface points and the associated normal vectors.
  13. Step: compute the robot Jacobians for selected joint position solutions.
  14. Step: compute the inverse of the robot Jacobian matrices and determine components $\widetilde{J}_T^{\dagger}$ and $\widetilde{J}_R^{\dagger}$ from the inverse matrices (see (50)).
  15. Step: compute the augmented inverse Jacobians by (55).

16. Step: compute the SVD of the augmented inverse Jacobians and generate ellipses (see Section 3.4).

4.1. Experimental Setup

The robotic setup is shown in Figure 2. We applied a collaborative robot arm UR5 since it is a widely adopted cobot in a low volume/high mix industry. It has a reach radius of 850 (mm); the kinematic parameters of the robot arm are available in reference [82]. The reference robot frame is shown by the red-green-blue axes located at the robot base. The MHP tool length was modeled as 284.5 (mm). The workpiece size was 250 × 250 × 75 (mm) and was placed with its center of the bottom face at the point [0, −525, −97] (mm) of the reference robot frame. In our numerical experiments, we applied two workpiece surface models, i.e., workpiece surface model No. 1 (WP1) and workpiece surface model No. 2 (WP2), which are shown in Figures 3 and 4, respectively. WP1 had a relatively monotone curvature, while WP2 had a more vivid curvature. The colormap shown on the workpiece surface is coded by its height.

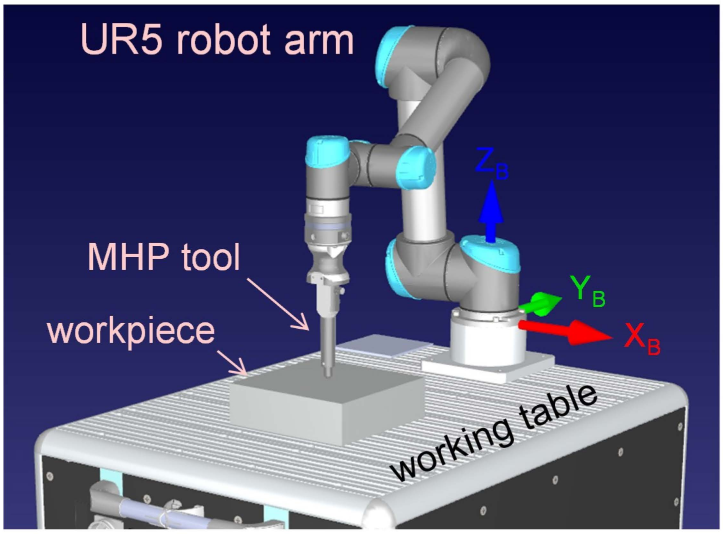


Figure 2. The flexible workstation for machine hammer peening with the collaborative robot UR5.

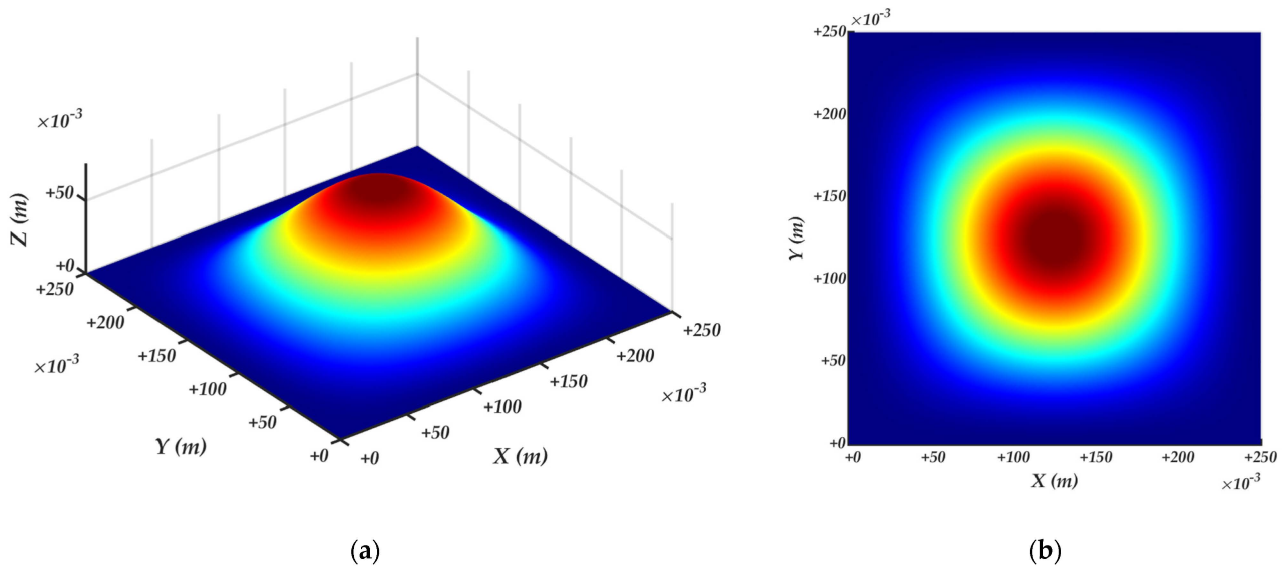
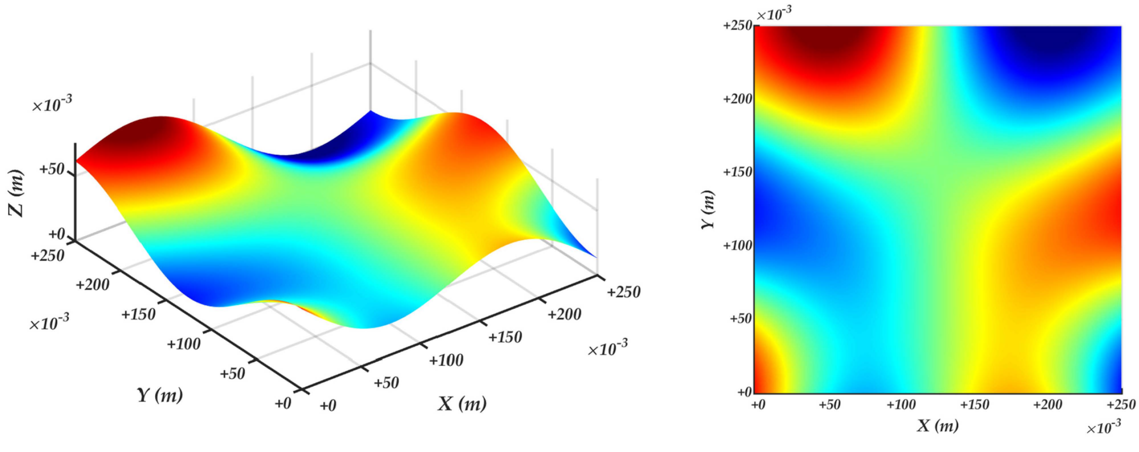


Figure 3. Workpiece surface model No. 1: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

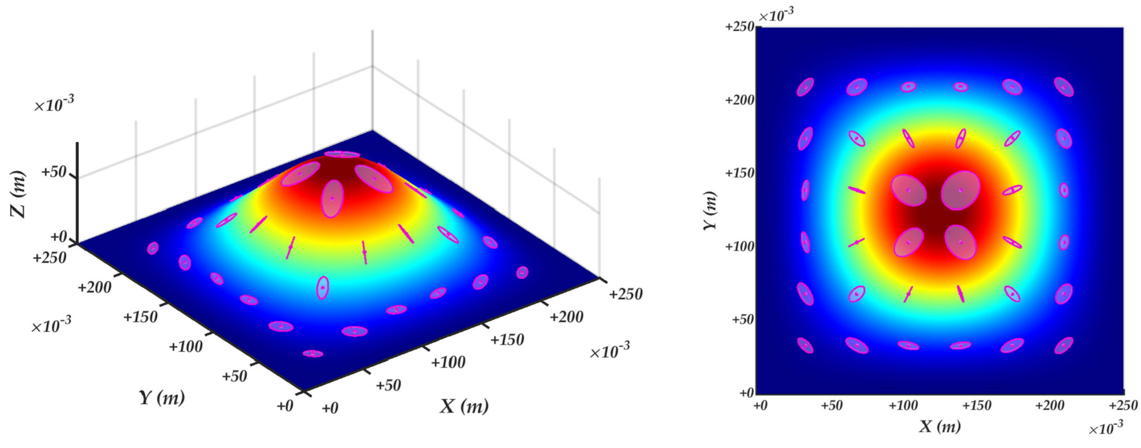


(**a**) (**b**)

Figure 4. Workpiece surface model No. 2: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

4.2. Surface Velocity Ellipsoids

The ellipsoids related to the velocity synchronization on the workpiece surface are presented in Figures 5–7. They are developed at the selected points on the workpiece's surface and represent the omnidirectional angular velocity (ω_C) performance of a particle moving along the path constrained on the workpiece surface under the condition $\|v_C\| \leq 1$. The surface velocity synchronization ellipsoids, which can be derived from (45)–(46), are shown firstly in Figure 5 for WP1 and Figure 6 for WP2. The ellipsoids are clearly of reduced dimension, such that they convert to ellipses and appear in the tangent planes attached to the selected points on the surface. The shown ellipses were all scaled by the factor of 0.00075. The angular velocity ellipses were of different sizes and eccentricity. Thus, the angular velocity performance depended both on the selected direction of the linear movement and on the geometry of the close neighborhood of the selected point on the workpiece surface. If a big ellipse similar to a circle was drawn, then a relatively high angular velocity would occur in all directions of the linear movement. In the case of an eccentric ellipse, the highest angular velocity would occur in the direction of the major ellipse axis and the lowest in the direction of the minor ellipse axis. The angular velocity performance depends highly on the surface slope (the surface gradient of the first order), its gradient (the surface gradient of the second order), and, thus, on the surface curvature.



(**a**) (**b**)

Figure 5. The surface velocity ellipses on WP1: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

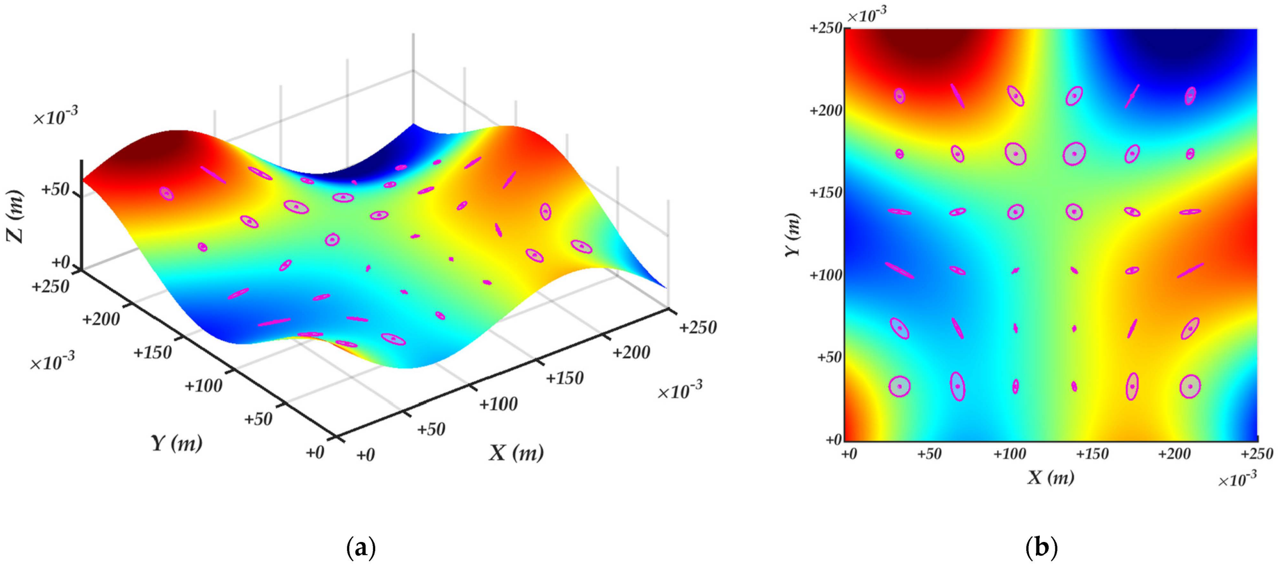


Figure 6. The surface velocity ellipses on WP2: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

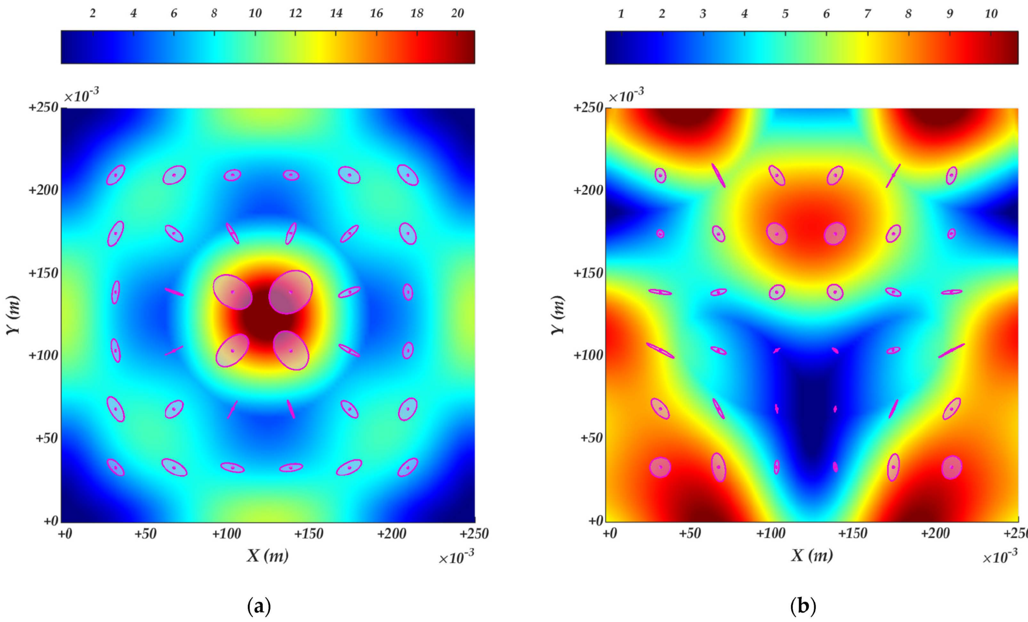
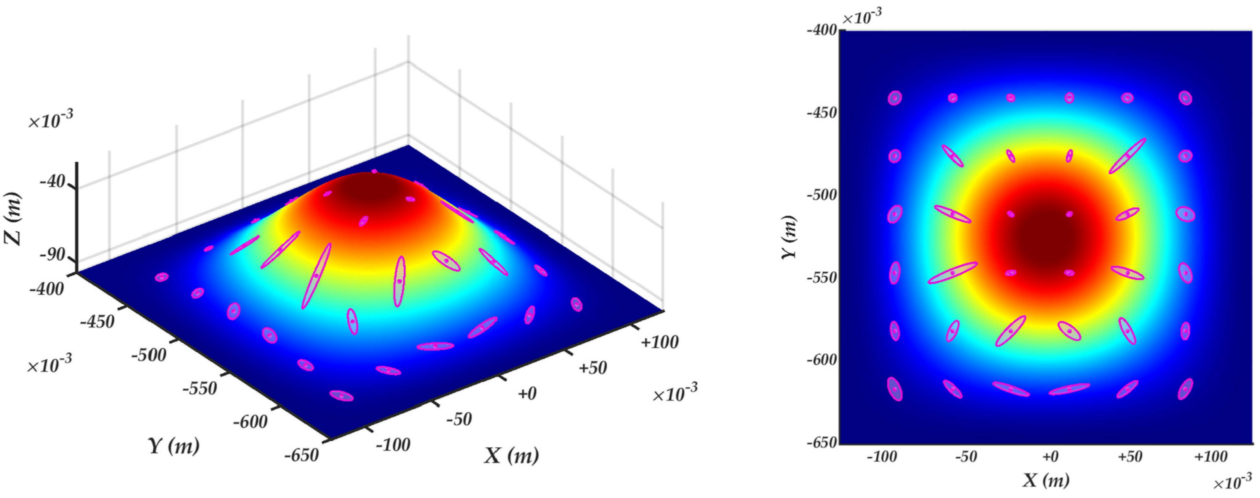


Figure 7. The surface velocity ellipses with mean surface curvature colormap code: (**a**) WP1; (**b**) WP2. The surface colormap is coded by surface mean curvature value (red color means high value, blue color means low value).

The same angular velocity ellipses are again shown in Figure 7. However, in this case, the surface colormap is coded by its mean curvature measure. A red/blue color means a high/low curvature measure value. It can be observed clearly that the drawn ellipses' characteristics correspond highly to the surface curvature measure at the selected points on both WP1 and WP2: at the surface points with a low value of the curvature measure, we can notice small ellipses, whereas at the surface points with a high value of the surface curvature measure, we can see big ellipses; in the case of the medium value curvature points, we can observe medium size ellipses, or ellipses with a longer major axis (and a medium average axis length), which again, match with the average curvature at the surface point. We can conclude that the angular velocity ellipses shown obviously correlate with the surface curvature characteristics.

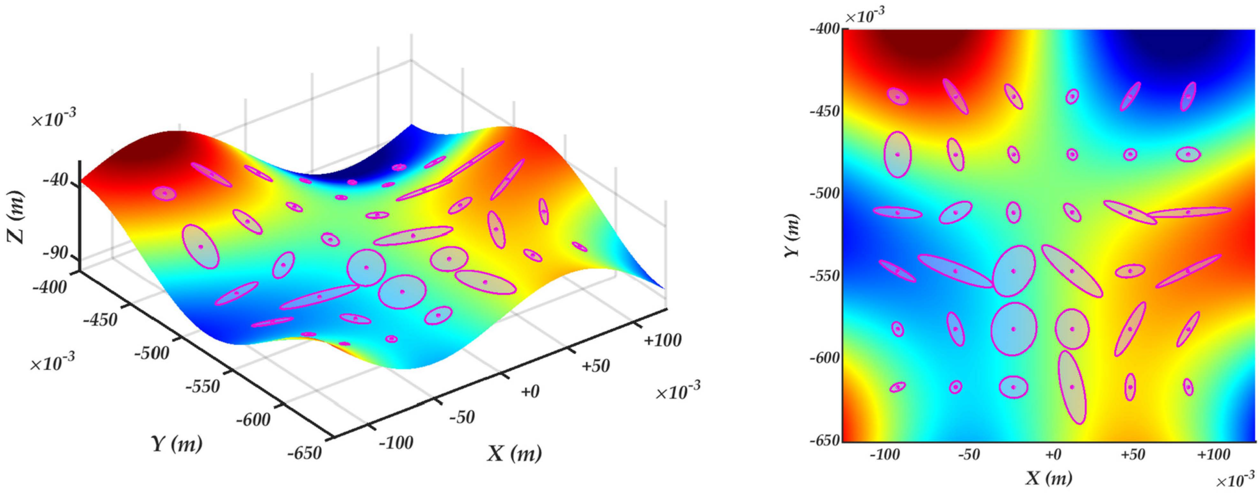
4.3. Robot Manipulability Ellipsoids of the Constrained Kinematics

The ellipsoids related to the robot's translational velocity along the path constrained on the workpiece with a curved surface and with the robot tool's orientation normal to the surface are presented in Figures 8–11. They express the robot's linear velocity performance at the positions in the robot's operational space determined by the selected points on the workpiece surface under the condition of unity Euclidian norm of the robot joint velocities $\|\dot{q}\| \leq 1$. The constrained manipulability ellipsoids, which can be derived from (54) and (55), are shown firstly in Figure 8 for WP1 and Figure 9 for WP2. The ellipsoids are again clearly of reduced dimensions, such that they convert to ellipses and appear in the tangent planes attached to the selected points on the surface. The shown ellipses were all scaled by a factor of 0.05.



(**a**) (**b**)

Figure 8. The velocity manipulability ellipses with robot kinematic constraints on WP1: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).



(**a**) (**b**)

Figure 9. The velocity manipulability ellipses with robot kinematic constraints on WP2: (**a**) perspective view; (**b**) top view. The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

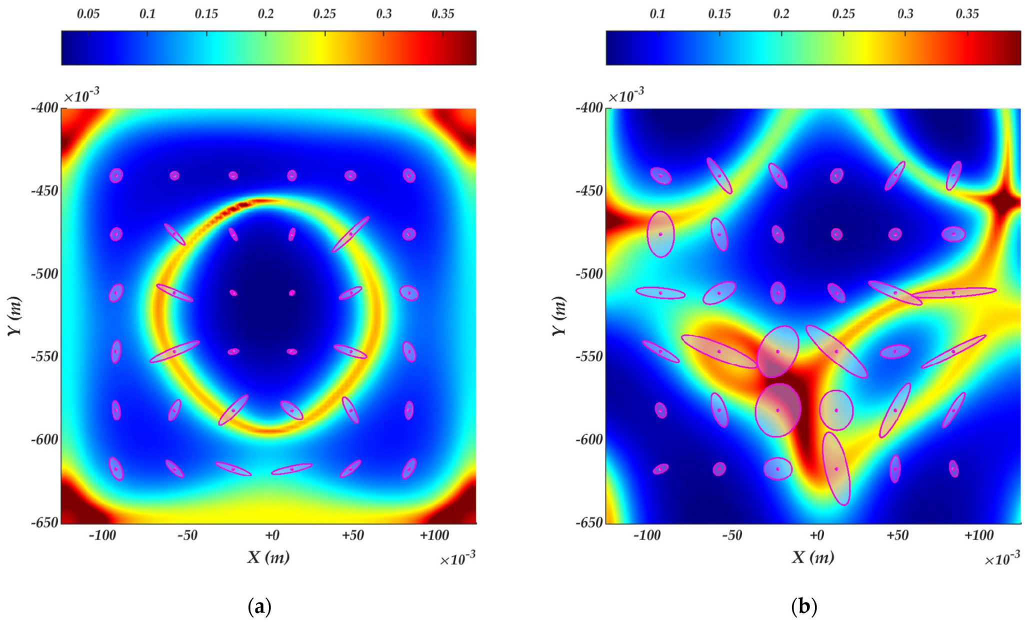


Figure 10. The velocity manipulability ellipses with mean velocity colormap code: (**a**) WP1; (**b**) WP2. The surface colormap is coded by mean (average) ellipse axes length value (red color means high value, blue color means low value).

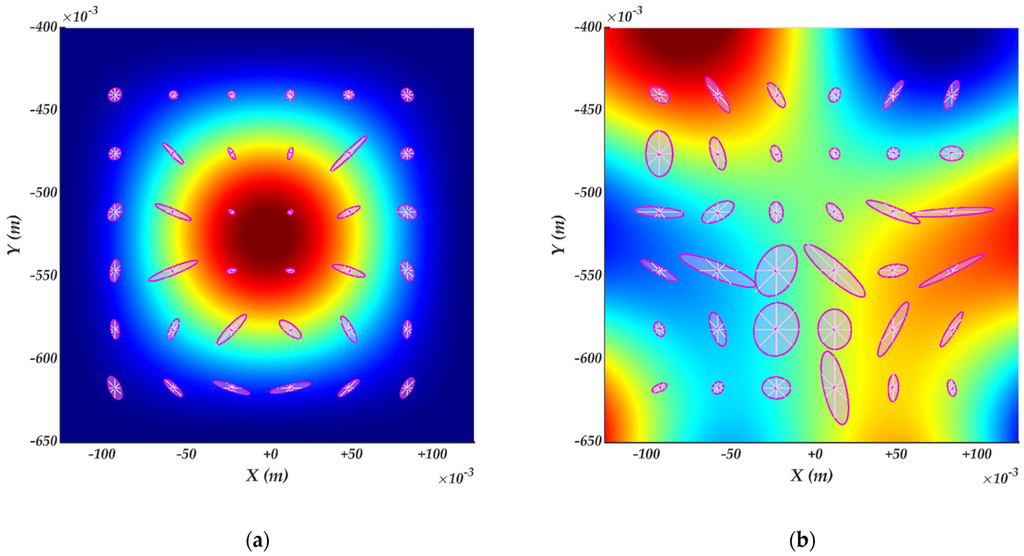


Figure 11. Verification of the manipulability velocity ellipses with robot kinematic constraints: (**a**) workpiece surface No. 1 (top view); (**b**) workpiece surface No. 2 (top view). The surface colormap is coded by surface point height value (red color means high value, blue color means low value).

At the bottom of WP1, all around Figure 8b, we can see ellipses with different sizes and orientations. Most of them are of medium size; some ellipses at the bottom side of the picture are longer and thinner, which determines a larger linear velocity performance in the direction of the ellipses' major axes and poorer in the direction of the minor axes; however, on the upper side of the picture, the ellipses are of small size, which is connected to poor linear velocity performance in all directions. Along the slope of WP1, we can see many rather long ellipses, with a major axis directed towards the top of the workpiece; it means that a straightway motion from the bottom to the top of the surface would be allowed at a relatively high speed. Finally, on the top of the workpiece, we can observe very small velocity ellipses since the surface in this area changed rapidly.

The velocity manipulability ellipsoids on WP2, with a more vivid curvature surface in Figure 9, are of different sizes, orientations, and widths. Their shape depends on the surface geometry at the specific point and the specific robot configuration required to provide the robot tool position and orientation that comply with the imposed surface constraints. Some big/medium size and only slightly eccentric ellipses, which can be observed mostly in the bottom half of the surface presented in Figure 9b, allow for high/medium linear omnidirectional velocity performance along the tangent planes attached to certain surface points. Thin ellipses on highs on the right side and on lows on the left side provide better velocity performance in certain directions determined by the ellipses' major axis. Many small size ellipses at different points all around the workpiece surface show only poor linear velocity performance.

A more detailed insight into the linear velocity performance is presented in Figure 10, where the surface colormap is coded by the mean (average) value of the ellipses' minor and major axes' lengths at all surface definition points. The red/blue color means roughly good/bad velocity manipulability. Figure 10a,b show the surface-constrained robot velocity manipulability for WP1/WP2. These characteristics can be used as a robot tool velocity regulation map on the surface at a certain workpiece location since they incorporate not only robot kinematics performance but also the workpiece surface curvature feature.

Finally, we verified the presented velocity ellipses for both WP1 and WP2. In the verification procedure, we simply calculated the robot tool linear velocity of maximal possible speed that can be achieved under condition $\|\dot{q}\| \leq 1$ during the motion along the path from the selected point to a certain neighboring point. In every ellipse, we selected eight neighboring points around the ellipse center point. In order to determine the maximum linear speed, we applied the following formula:

$$V_{\max} = \frac{1}{\left\|\widetilde{J}_T^{\dagger} u_T + h^{-1}\widetilde{J}_R^{\dagger} u_R\right\|_2}, \tag{68}$$

where u_T and u_R are linear and angular directions along the path, whereas the parameter h is related to the surface shape/curvature in our case and can be calculated as $h = s/\varphi$, where s and φ stand for linear path length and rotational path length (angle), respectively. The formula above is similar to the DTF method formula [30]; however, please note that the DTF method employs the infinity norm in the denominator, whereas we used the Euclidean L_2 norm. The verification results for WP1/WP2 are shown in Figure 11a,b. Here, the surface colormap is coded by the surface height. Again, we can see the before-mentioned manipulability ellipses, as in Figures 8 and 9. In addition, there are the verification linear velocity vectors drawn in the bright pale pink color. They were all scaled by the factor of 0.05, which is of the same value as the manipulability ellipses' scaling factor. We can observe that the vector lines drawn from the ellipses' center point end at the ellipses' edge. Thus, they correspond to the velocities, which could be determined from the manipulability ellipses in the verification directions and, therefore, confirm the correctness of the presented manipulability ellipses of the robot kinematics imposed by the workpiece surface constraints.

5. Discussion

The main results obtained can be summed up as follows. The proposed method effectively avoids the problems related to the inhomogeneous Jacobian in the manipulability concept in a non-arbitrary way. The derived augmented inverse Jacobian matrix with incorporated workpiece surface differential constraints virtually reduces the task space to the translational subspace. Furthermore, robot motion planning can be performed in the two-dimensional surface tangent planes of a workpiece. To show the viability of the proposed augmented inverse Jacobian, we utilize well-known manipulability ellipsoids. However, in our case, the ellipsoids were reduced to two-dimensional ellipses, each lying in its own surface tangent plane. The resulting ellipses were clearly verified. They enable obtaining optimal linear velocity in terms of magnitude and direction for machining paths embedded on the workpiece surface under the assumption that the robot tool orientation is kept normal to the workpiece surface.

However, to obtain the augmented inverse Jacobian matrix, we need extensive mathematical computation/processing of a workpiece surface to provide the required differential geometry characteristics. It is necessary to compute the first-order and the second-order partial derivatives of the workpiece surface. Thus, the accurate workpiece definition must be available for this purpose. In this paper, we provided grid-based surface models and standard mathematical processing approaches to compute the required surface derivatives. Note that the processing methodology may be properly adapted for use with practical CAD formats of a workpiece. Nevertheless, the differential geometry characteristics of a workpiece should be computed as accurately as possible since they significantly affect the precision of the proposed augmented inverse Jacobian matrix and, consequently, the related manipulability analysis as well.

6. Conclusions

In this paper, we discussed the robot kinematics' capability and velocity performance in the case of workpiece machining, where the robot tool must follow a path embedded on the surface at the normal orientation. In the presented study, we considered a workpiece with complex surface geometry to derive the robot's constrained kinematics description. The derived augmented Jacobian kinematics matrix with incorporated workpiece surface constraints was, in the following, used for the ellipsoid-based manipulability analysis. We have shown that, in this case, the general non-homogeneous six-dimensional manipulability ellipsoids, which mix different units of linear velocity and angular velocity, can be reduced to the two-dimensional ellipses in the homogeneous translational subspace. Thus, the problem related to the inhomogeneous Jacobian was effectively avoided in a non-arbitrary way. The results from the theoretical development have been proven by numerical experiments. In the future, the presented approach of the surface-based constrained manipulability can be employed in the kinematic optimization of the robot-workpiece position and planning of the optimal robot machining path with the objective of achieving maximum Cartesian speed concerning robot joints' limits. Furthermore, future experimental validation may include real-world manufacturing examples in a real robotic cell with a collaborative robot.

Funding: This research was funded by the Slovenian Research Agency (ARRS) under a Grant for the Research Program P2–0028 and in part by a Project of the Republic of Slovenia and the European Union from the European Regional Development Fund, ROBOTOOL-1, Grant number OP20.03540.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. International Federation of Robotics: World Robotics Industrial Robots. Available online: https://ifr.org/worldrobotics (accessed on 8 August 2023).
2. Magrini, E.; Ferraguti, F.; Ronga, A.J.; Pini, F.; De Luca, A.; Leali, F. Human-robot coexistence and interaction in open industrial cells. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101846. [CrossRef]
3. Taesi, C.; Aggogeri, F.; Pellegrini, N. COBOT Applications: Recent Advances and Challenges. *Robotics* **2023**, *12*, 79. [CrossRef]
4. Gajšek, B.; Stradovnik, S.; Hace, A. Sustainable Move towards Flexible, Robotic, Human-Involving Workplace. *Sustainability* **2020**, *12*, 6590. [CrossRef]
5. Weidemann, C.; Mandischer, N.; van Kerkom, F.; Corves, B.; Hüsing, M.; Kraus, T.; Garus, C. Literature Review on Recent Trends and Perspectives of Collaborative Robotics in Work 4.0. *Robotics* **2023**, *12*, 84. [CrossRef]
6. Aivaliotis, P.; Aivaliotis, S.; Gkournelos, C.; Kokkalis, K.; Michalos, G.; Makris, S. Power and force limiting on industrial robots for human-robot collaboration. *Robot. Comput.-Integr. Manuf.* **2019**, *59*, 346–360. [CrossRef]
7. Ji, W.; Wang, L. Industrial robotic machining: A review. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 1239–1255. [CrossRef]
8. Qu, X.; Wan, M.; Shen, C.-J.; Zhang, W.-H. Profile error-oriented optimization of the feed direction and posture of the end-effector in robotic free-form milling. *Robot. Comput.-Integr. Manuf.* **2023**, *83*, 102580. [CrossRef]

9. Onstein, I.F.; Bjerkeng, M.; Martinsen, K. Automated Tool Trajectory Generation for Robotized Deburring of Cast Parts Based on 3D Scans. *Procedia CIRP* **2023**, *118*, 507–512. [CrossRef]
10. Bogue, R. Finishing robots: A review of technologies and applications. *Ind. Robot. Int. J.* **2009**, *36*, 6–12. [CrossRef]
11. Yang, J.; Chen, H.; Qi, R.; Ding, H.; Yin, Y. A novel approach to robotic grinding guaranteeing profile accuracy using rigid-flexible coupling force control for free-formed surfaces. *CIRP Ann.* **2023**, *72*, 313–316. [CrossRef]
12. Kharidege, A.; Ting, D.T.; Yajun, Z. A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot. *Int. J. Adv. Manuf. Technol.* **2017**, *93*, 3921–3934. [CrossRef]
13. Ke, X.; Yu, Y.; Li, K.; Wang, T.; Zhong, B.; Wang, Z.; Kong, L.; Guo, J.; Huang, L.; Idir, M.; et al. Review on robot-assisted polishing: Status and future trends. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102482. [CrossRef]
14. Bleicher, F.; Lechner, C.; Habersohn, C.; Kozeschnik, E.; Adjassoho, B.; Kaminski, H. Mechanism of surface modification using machine hammer peening technology. *CIRP Ann.* **2012**, *61*, 375–378. [CrossRef]
15. Hagen, L.; Baumann, J.; Heidelmann, M.; Biermann, D.; Tillmann, W. Machine Hammer Peening and Its Effect on the Surface Integrity of Arc-Sprayed WC-W2C-FeCMnSi Coatings. *Coatings* **2023**, *13*, 1563. [CrossRef]
16. Makulavičius, M.; Petkevičius, S.; Rožėnė, J.; Dzedzickis, A.; Bučinskas, V. Industrial Robots in Mechanical Machining: Perspectives and Limitations. *Robotics* **2023**, *12*, 160. [CrossRef]
17. Schulze, V.; Bleicher, F.; Groche, P.; Guo, Y.B.; Pyun, Y.S. Surface modification by machine hammer peening and burnishing. *CIRP Ann.* **2016**, *65*, 809–832. [CrossRef]
18. Hähnel, S.; Pini, F.; Leali, F.; Dambon, O.; Bergs, T.; Bletek, T. Reconfigurable Robotic Solution for Effective Finishing of Complex Surfaces. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; pp. 1285–1290.
19. KUKA. Surface processing with a KR QUANTEC and KUKA.CNC. Available online: https://www.kuka.com/en-de/industries/solutions-database/2016/07/solution-robotics-sematek (accessed on 9 August 2023).
20. Robotmaster. Offline Programming Software Is Single Solution for Robotic Hammer Peening of Automotive Stamping Dies. Available online: https://www.roboticstomorrow.com/article/2017/12/offline-programming-software-is-single-solution-for-robotic-hammer-peening-of-automotive-stamping-dies/11173 (accessed on 9 August 2023).
21. Weingartshofer, T.; Bischof, B.; Meiringer, M.; Hartl-Nesic, C.; Kugi, A. Optimization-based path planning framework for industrial manufacturing processes with complex continuous paths. *Robot. Comput.-Integr. Manuf.* **2023**, *82*, 102516. [CrossRef]
22. Gasparetto, A.; Boscariol, P.; Lanzutti, A.; Vidoni, R. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*; Springer: Cham, Swittherland, 2015; pp. 3–27.
23. Nektarios, A.; Aspragathos, N.A. Optimal location of a general position and orientation end-effector's path relative to manipulator's base, considering velocity performance. *Robot. Comput.-Integr. Manuf.* **2010**, *26*, 162–173. [CrossRef]
24. Doan, N.C.N.; Lin, W. Optimal robot placement with consideration of redundancy problem for wrist-partitioned 6R articulated robots. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 233–242. [CrossRef]
25. Xia, Y.; Zhao, L.; Liu, Z.; Zhu, K. Optimization of Machining Position for Wheel Hub Polishing Robot Based on the Kinematic Index. In Proceedings of the 2022 8th International Conference on Control, Automation and Robotics (ICCAR), Xiamen, China, 8–10 April 2022; pp. 155–160.
26. Weingartshofer, T.; Hartl-Nesic, C.; Kugi, A. Optimal TCP and Robot Base Placement for a Set of Complex Continuous Paths. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 9659–9665.
27. Malhan, R.K.; Kabir, A.M.; Shah, B.; Gupta, S.K. Identifying Feasible Workpiece Placement with Respect to Redundant Manipulator for Complex Manufacturing Tasks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5585–5591.
28. Malhan, R.K.; Shembekar, A.V.; Kabir, A.M.; Bhatt, P.M.; Shah, B.; Zanio, S.; Nutt, S.; Gupta, S.K. Automated planning for robotic layup of composite prepreg. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 102020. [CrossRef]
29. Balci, B.; Donovan, J.; Roberts, J.; Corke, P. Optimal Workpiece Placement Based on Robot Reach, Manipulability and Joint Torques. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 12302–12308.
30. Stradovnik, S.; Hace, A. Task-Oriented Evaluation of the Feasible Kinematic Directional Capabilities for Robot Machining. *Sensors* **2022**, *22*, 4267. [CrossRef] [PubMed]
31. Sun, Y.; Jia, J.; Xu, J.; Chen, M.; Niu, J. Path, feedrate and trajectory planning for free-form surface machining: A state-of-the-art review. *Chin. J. Aeronaut.* **2022**, *35*, 12–29. [CrossRef]
32. Amersdorfer, M.; Meurer, T. Equidistant Tool Path and Cartesian Trajectory Planning for Robotic Machining of Curved Freeform Surfaces. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3311–3323. [CrossRef]
33. Wen, Y.; Jaeger, D.J.; Pagilla, P.R. Uniform Coverage Tool Path Generation for Robotic Surface Finishing of Curved Surfaces. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4931–4938. [CrossRef]
34. Xiao, M.; Ding, Y.; Yang, G. A Model-Based Trajectory Planning Method for Robotic Polishing of Complex Surfaces. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 2890–2903. [CrossRef]

35. Liao, Z.-Y.; Li, J.-R.; Xie, H.-L.; Wang, Q.-H.; Zhou, X.-F. Region-based toolpath generation for robotic milling of freeform surfaces with stiffness optimization. *Robot. Comput.-Integr. Manuf.* **2020**, *64*, 101953. [CrossRef]
36. Xie, H.-L.; Li, J.-R.; Liao, Z.-Y.; Wang, Q.-H.; Zhou, X.-F. A robotic belt grinding approach based on easy-to-grind region partitioning. *J. Manuf. Process.* **2020**, *56*, 830–844. [CrossRef]
37. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*; Cambridge University Press: Cambridge, UK, 2017.
38. Altuzarra, O.; Pinto, C.; Sandru, B.; Hernandez, A. Optimal Dimensioning for Parallel Manipulators: Workspace, Dexterity, and Energy. *J. Mech. Des.* **2011**, *133*, 041007. [CrossRef]
39. Kelaiaia, R.; Chemori, A.; Brahmia, A.; Kerboua, A.; Zaatri, A.; Company, O. Optimal dimensional design of parallel manipulators with an illustrative case study: A review. *Mech. Mach. Theory* **2023**, *188*, 105390. [CrossRef]
40. Zha, X.F. Optimal pose trajectory planning for robot manipulators. *Mech. Mach. Theory* **2002**, *37*, 1063–1086. [CrossRef]
41. Lu, X.; Jia, Y. Trajectory Planning of Free-Floating Space Manipulators With Spacecraft Attitude Stabilization and Manipulability Optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7346–7362. [CrossRef]
42. Pardi, T.; Ortenzi, V.; Fairbairn, C.; Pipe, T.; Esfahani, A.M.G.; Stolkin, R. Planning Maximum-Manipulability Cutting Paths. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1999–2006. [CrossRef]
43. Vahrenkamp, N.; Asfour, T.; Metta, G.; Sandini, G.; Dillmann, R. Manipulability analysis. In Proceedings of the 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Osaka, Japan, 29 November–1 December 2012; pp. 568–573.
44. Vahrenkamp, N.; Asfour, T. Representing the robot's workspace through constrained manipulability analysis. *Auton. Robot.* **2015**, *38*, 17–30. [CrossRef]
45. Kaden, S.; Thomas, U. Maximizing Robot Manipulability along Paths in Collision-free Motion Planning. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 105–110.
46. Dufour, K.; Suleiman, W. On Maximizing Manipulability Index while Solving a Kinematics Task. *J. Intell. Robot. Syst.* **2020**, *100*, 3–13. [CrossRef]
47. Marić, F.; Limoyo, O.; Petrović, L.; Ablett, T.; Petrović, I.; Kelly, J. Fast Manipulability Maximization Using Continuous-Time Trajectory optimization. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 8258–8264.
48. Shen, H.; Xie, W.F.; Tang, J.; Zhou, T. Adaptive Manipulability-Based Path Planning Strategy for Industrial Robot Manipulators. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 1742–1753. [CrossRef]
49. Corves, B.; Shahidi, A. Kinematic Graph for Motion Planning of Robotic Manipulators. *Robotics* **2022**, *11*, 105. [CrossRef]
50. Yoshikawa, T. Manipulability of Robotic Mechanisms. *Int. J. Robot. Res.* **1985**, *4*, 3–9. [CrossRef]
51. Klein, C.A.; Blaho, B.E. Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators. *Int. J. Robot. Res.* **1987**, *6*, 72–83. [CrossRef]
52. Schwartz, E.; Manseur, R.; Doty, K. Noncommensurate systems in robotics. *Int. J. Robot. Automat.* **2002**, *17*, 86–92.
53. Melchiorri, C.; Chiaccio, P.; Chiaverini, S.; Sciavicco, L.; Siciliano, B. Comments on "Global task space manipulability ellipsoids for multiple-arm systems" and further considerations" [with reply] P. Chiacchio; et al. *IEEE Trans. Robot. Autom.* **1993**, *9*, 232–236. [CrossRef]
54. Gosselin, C.M. The optimum design of robotic manipulators using dexterity indices. *Robot. Auton. Syst.* **1992**, *9*, 213–226. [CrossRef]
55. Cardou, P.; Bouchard, S.; Gosselin, C. Kinematic-Sensitivity Indices for Dimensionally Nonhomogeneous Jacobian Matrices. *IEEE Trans. Robot.* **2010**, *26*, 166–173. [CrossRef]
56. Dubey, R.; Luh, J.Y.S. Redundant robot control using task based performance measures. *J. Robot. Syst.* **1988**, *5*, 409–432. [CrossRef]
57. Chiu, S.L. Task Compatibility of Manipulator Postures. *Int. J. Robot. Res.* **1988**, *7*, 13–21. [CrossRef]
58. Wang, Y.; Wang, L. Reactive task-oriented redundancy resolution using constraint-based programming. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 5689–5694.
59. Doty, K.L.; Melchiorri, C.; Schwartz, E.M.; Bonivento, C. Robot manipulability. *IEEE Trans. Robot. Autom.* **1995**, *11*, 462–468. [CrossRef]
60. Staffetti, E.; Bruyninckx, H.; De Schutter, J. On the Invariance of Manipulability Indices. In *Advances in Robot Kinematics: Theory and Applications*; Lenarčič, J., Thomas, F., Eds.; Springer: Dordrecht, The Netherlands, 2002; pp. 57–66.
61. Yoshikawa, T. Translational and rotational manipulability of robotic manipulators. In Proceedings of the 1991 International Conference on Industrial Electronics, Control and Instrumentation (IECON '91), Kobe, Japan, 28 October–1 November 1991; pp. 1170–1175.
62. Mansouri, I.; Ouali, M. The power manipulability—A new homogeneous performance index of robot manipulators. *Robot. Comput.-Integr. Manuf.* **2011**, *27*, 434–449. [CrossRef]
63. Mansfeld, N.; Keppler, M.; Haddadin, S. Speed Gain in Elastic Joint Robots: An Energy Conversion-Based Approach. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4600–4607. [CrossRef]
64. Kokkinis, T.; Paden, B. Kinetostatic performance limits of cooperating robot manipulators using force-velocity polytopes. In Proceedings of the ASME Winter Annual Meeting-Robotics Research, vol. 14, San Francisco, CA, USA, 8–9 November 1989; Volume 14, pp. 151–155.

65. Lee, J. A study on the manipulability measures for robot manipulators. In Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems, Innovative Robotics for Real-World Applications, IROS '97, Grenoble, France, 7–11 September 1997; pp. 1458–1465.
66. Finotello, R.; Grasso, T.; Rossi, G.; Terribile, A. Computation of kinetostatic performances of robot manipulators with polytopes. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), Leuven, Belgium, 20–20 May 1998; Volume 3244, pp. 3241–3246.
67. Long, P.; Padir, T. Evaluating Robot Manipulability in Constrained Environments by Velocity Polytope Reduction. In Proceedings of the 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 6–9 November 2018; pp. 1–9.
68. Long, P.; Padir, T. Constrained Manipulability for Humanoid Robots Using Velocity Polytopes. *Int. J. Humanoid Robot.* **2019**, *17*, 1950037. [CrossRef]
69. Kang, L.; Kim, S.H.; Kim, W.; Yi, B.J. Review of Dimension Inhomogeneity in Robotics. In Proceedings of the 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Republic of Korea, 24–27 June 2019; pp. 143–148.
70. Patrikalakis, N.M.; Maekawa, T. *Shape Interrogation for Computer Aided Design and Manufacturing*; Springer: Berlin/Heidelberg, Germany, 2002.
71. Kobayashi, S. *Differential Geometry of Curves and Surfaces*; Springer: Singapore, 2019.
72. Hertrich-Jeromin, U. Differential Geometry (Lecture Notes). 2019. Available online: https://www.geometrie.tuwien.ac.at/hertrich-jeromin/pub/dg.pdf (accessed on 26 November 2023).
73. Strang, G. *Introduction to Linear Algebra*, 5th ed.; Cambridge Press: Wellesley, MA, USA, 2016.
74. Nakamura, Y. *Advanced Robotics: Redundancy and Optimization*; Addison-Wesley: Boston, MA, USA, 1991.
75. Patel, S.; Sobh, T. Manipulator Performance Measures—A Comprehensive Literature Survey. *J. Intell. Robot. Syst.* **2014**, *77*, 547–570. [CrossRef]
76. Baksalary, J.K.; Baksalary, O.M. Particular formulae for the Moore–Penrose inverse of a columnwise partitioned matrix. *Linear Algebra Its Appl.* **2007**, *421*, 16–23. [CrossRef]
77. Baksalary, O.M.; Trenkler, G. On formulae for the Moore–Penrose inverse of a columnwise partitioned matrix. *Appl. Math. Comput.* **2021**, *403*, 125913. [CrossRef]
78. Moulianitis, V.; Katrantzis, E.; Stravopodis, N.; Aspragathos, N. A Comparative Study of Three Manipulator Performance Measures. In *Advances in Service and Industrial Robotics, Proceedings of the 26th International Conference on Robotics in Alpe-Adria-Danube Region, RAAD 2017, Torino, Italy, 21–23 June 2017*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 19–27.
79. Maciejewski, A.A.; Klein, C.A. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments. *Int. J. Robot. Res.* **1985**, *4*, 109–117. [CrossRef]
80. Piziak, R.; Odell, P.L. *Matrix Theory—From Generalized Inverses to Jordan Form*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2007.
81. MathWorks: MATLAB. Available online: https://ch.mathworks.com/products/matlab.html (accessed on 15 November 2023).
82. Universal Robots: DH Parameters for Calculations of Kinematics and Dynamics. Available online: https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/ (accessed on 15 November 2023).