Repeatable approximation of the Jacobian pseudo-inverse

Krzysztof Tchoń and Mariusz Janiak

Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
ul. Janiszewskiego 11/17,
50–372 Wrocław, Poland
email: krzysztof.tchon|mariusz.janiak@pwr.wroc.pl

Abstract

For redundant robot kinematics we construct an extended Jacobian inverse that approximates the Jacobian pseudo-inverse. Our construction relies on the approximation of codistributions associated with the Jacobian pseudo-inverse by an integrable codistribution, spanned by differentials of certain functions that serve as components of the augmenting kinematics map in the extended Jacobian inverse. The procedure is applied to the kinematics of a 7 degree-of-freedom manipulator POLYCRANK, and to the kinematics of a mobile robot in the chained-form with degree of redundancy 2.

Key words: Redundant robot kinematics, Jacobian pseudo-inverse, extended Jacobian inverse, codistribution, approximation

1. Introduction

Suppose that the map \( k : R^n \rightarrow R^m \), \( y = k(x) \), where \( m < n \), denotes a coordinate representation of the redundant forward kinematics of a robot. This can be either the kinematics of a manipulator or a finite-dimensional representation of the kinematics of a mobile robot. Given a task space point \( y_d \), the inverse kinematics problem consists in determining a joint position \( x_d \) such that \( k(x_d) = y_d \). The inverse kinematics problem can be solved by means of the continuation method, [1, 2, 3], in accordance with the following procedure. We begin by choosing an initial configuration \( x_0 \), then we take a smooth curve \( x(\theta) \), \( \theta \in R \), such that \( x(0) = x_0 \), and define a task space error...
\[ e(\theta) = k(x(\theta)) - y_d \text{ along this curve. The curve is supposed to provide a desirable (e.g. exponential) convergence of the error to 0, with a rate } \gamma > 0, \text{ i.e.} \]
\[ \frac{d e(\theta)}{d \theta} = -\gamma e(\theta). \]

A differentiation of the error \( e(\theta) \) leads to the differential equation
\[ J(x) \frac{dx}{d \theta} = -\gamma (k(x) - y_d) \]

for the curve \( x(\theta) \), that is sometimes referred to as the Ważewski-Davidenko equation [4]. For every right inverse \( J^#(x) \) of the Jacobian \( J(x) \) this equation produces an inverse kinematics algorithm in the form of a dynamic system
\[ \frac{dx}{d \theta} = -\gamma J^#(x)(k(x) - y_d). \] (1)

A solution of the problem is obtained as the limit \( x_d = \lim_{\theta \to +\infty} x(\theta) \) of the trajectory of (1).

Geometrically, the columns of the matrix \( J^#(x) \) may be treated as vector fields that span an \( m \)-dimensional distribution \[ \mathcal{D} = \text{span}_{\mathbb{R}^s} \{J^1^#(x), \ldots, J^m^#(x)\} \]
associated with (1). The most often used right inverse of the Jacobian is the Jacobian pseudo-inverse (the Moore-Penrose inverse of the Jacobian), defined as \( J^p^p(x) = J^T(x) \left(J(x)J^T(x)\right)^{-1} \), whose associated distribution is generated by the columns of the Jacobian transpose \( J^T(x) \). Another class of inverse kinematics algorithms employ the concept of the Jacobian extension. Given the kinematics \( k(x) \) with degree of redundancy \( r = n - m \), we add an augmenting kinematics map \( h: \mathbb{R}^n \to \mathbb{R}^r \), and introduce the extended Jacobian [6]
\[ \bar{J}(x) = \begin{bmatrix} J(x) \\ \frac{\partial h(x)}{\partial x} \end{bmatrix}. \] (3)

Outside singularities of \( \bar{J}(x) \) the extended Jacobian inverse is defined as
\[ J^E_E(x) = \bar{J}(x)^{-1}|_{m \text{ first columns}}. \] (4)

By definition, the extended Jacobian inverse has two properties: it is a right inverse of the Jacobian, \( J(x)J^E_E(x) = I_m \), and it is annihilated by the augmenting Jacobian,
\[ \frac{\partial h_i(x)}{\partial x} J^H(x) = 0. \] Actually, the last identity means that the vector fields \( J^H_1(x), \ldots, J^H_m(x) \) are orthogonal to the differentials \( dh_1(x), \ldots, dh_r(x) \).

Let \( \omega_1(x), \ldots, \omega_r(x) \) denote a collection of independent one-forms. Then, an object

\[ \Omega = \text{span}_{C^\infty(\mathbb{R}^n)} \{ \omega_1(x), \ldots, \omega_r(x) \} \]

is called a codistribution [5]. By the annihilation property the distribution associated with the extended Jacobian inverse is annihilated by the codistribution spanned by the differentials \( dh_1(x), \ldots, dh_r(x) \).

The Jacobian pseudo-inverse algorithm distinguishes by a fast, quadratic convergence, whereas the extended Jacobian algorithm is repeatable. We remind that the repeatability of an inverse kinematics algorithm means that every closed path in the task space is transformed by the algorithm into a closed path in the configuration space. Repeatable control strategies become most advantageous when the robot carries out a cyclic sequence of tasks. It is well known that the Jacobian pseudo-inverse inverse kinematics algorithm for stationary manipulators is not repeatable [7]. A similar result for mobile manipulators can be found in [8]. Necessary and sufficient conditions for repeatability of inverse kinematics algorithms for stationary manipulators have been derived by [9], and generalized to mobile manipulators by [8]. These conditions require integrability of the associated distribution. For the reason that the extended Jacobian inverse has the annihilation property, its associated distribution is involutive (so integrable), therefore the extended Jacobian inverse kinematics algorithm is repeatable by design.

An idea of making a fusion of convergence and repeatability properties inherent in these two inverse kinematics algorithms consists in defining a repeatable (extended Jacobian) inverse that would approximate in an optimal way the Jacobian pseudo-inverse. This approximation problem has been formulated in terms of the minimization of an error functional in [10, 11]. It can be checked that the optimality conditions for this approximation, provided by the methods of the calculus of variations, are rather complex. Furthermore, for the set of repeatable inverses is topologically small (of infinite codimension), such an approximation presents a sort of ill-posed problem.

The objective of this paper is to set forth a new approach to the synthesis of the ex-
tended Jacobian inverse kinematics algorithms that approximate the Jacobian pseudo-inverse. Instead of the variational approach, we prefer a differential geometric formulation of the approximation problem, that relies on the method of approximation of codistributions expounded in [12]. Given a codistribution, the method provides an integrable codistribution that coincides with the original codistribution on certain sub-manifolds of the configuration space and along the integral curve of a specific vector field associated with a homotopy map. If the kinematics have the redundancy degree \( r \), these codistributions will agree on some \((r + 1)\)-dimensional submanifolds of the configuration space, referred to as pages [12]. The resulting codistribution is spanned by differentials of certain functions that are further used as the components of the augmenting kinematics map. Mathematically, the augmenting kinematics map results from solving a collection of \( r \) identical partial differential equations by means of the method of characteristics. In our recent paper [4] we have illustrated this approach with simple examples of kinematics with redundancy degree 1. Here we shall formulate mathematical foundations of the approach, and show its applications to a 7 degree-of-freedom manipulator POLYCRANK, and to the chained-form mobile robot kinematics of redundancy 2. In the latter case the augmenting kinematics map has been found analytically, in the former we need to resort to numerical computations.

The composition of this paper is the following. In Section 2 presents mathematical foundations of our approach. Section 3 specifies the approach to the POLYCRANK manipulator. In section 4 the approach is applied to a chained-form system. Section 5 illustrates the performance of the extended Jacobian algorithms vs. the Jacobian pseudo-inverse with computer simulations. The paper concludes with Section 6.

2. Basic concepts

As we have already stated, the distribution associated with Jacobian pseudo-inverse \( J^\#P(x) = J^T(x)(J(x)J^T(x))^{-1} \) is spanned by the columns of \( J^T(x) \). This being so, let us define a codistribution

\[
\Omega_P = \text{span}_{C^\infty(R^n)} \left\{ \omega_1(x), \ldots, \omega_r(x) \right\}
\]
spanned by one-forms that annihilate this distribution, so $\omega_i(x)J^T(x) = 0$, $i = 1, \ldots, r$, the number $r$ denoting the redundancy degree of the kinematics. Our objective consists in designing an extended Jacobian inverse determined by the augmenting kinematics map $h(x) = (h_1(x), \ldots, h_r(x))$, such that the codistribution
\[ \Omega_E = \text{span}_{\mathbb{R}^r} \{ dh_1(x), \ldots, dh_r(x) \}, \]
spanned by the differentials of the components of $h$, agrees with the codistribution (5) on prespecified submanifolds of the configuration space and along a prespecified direction of motion in the configuration space, as proposed in [12]. More specifically, let us introduce into the configuration space $\mathbb{R}^n$ a foliation by $r$-dimensional leaves $E_\alpha$, with a distinguished zero-leaf $E_0$. Then we define a homotopy $\Phi_t : \mathbb{R}^n \to \mathbb{R}^n$, with parameter $t \in [0, 1]$, preserving these leaves (i.e. if $x^1, x^2$ belong to a leaf then for every $t$ the images $\Phi_t(x^1), \Phi_t(x^2)$ will stay in the same leaf), moreover, we assume that $\Phi_1 = id_{E_0}$, $\Phi_0 : \mathbb{R}^n \to E_0$, and $\Phi_s \circ \Phi_t = \Phi_{st}$, and let
\[ X(x) = \left. \frac{d\Phi_t(x)}{dt} \right|_{t=1}. \]
Relying on the definition, we compute
\[ \left. \frac{d\Phi_{st}(x)}{ds} \right|_{s=1} = \frac{d\Phi_t \circ \Phi_s(x)}{ds} \bigg|_{s=1} = X(\Phi_t(x)). \]
On the other hand, we have
\[ \left. \frac{d\Phi_s(x)}{ds} \right|_{s=1} = \left. \frac{d\Phi_u(x)}{du} \right|_{s=1} \frac{du}{ds} = \left. \frac{d\Phi_t(x)}{dt} \right|_{s=1}, \]
what leads to the conclusion
\[ i \left. \frac{d\Phi_t(x)}{dt} \right|_{s=1} = X(\Phi_t(x)). \]
For further mathematical developments, we shall partition the configuration coordinates as $x = (y, z)$, where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^r$, in such a way that $y$ is transverse to the leaf and $z$ refers to the leaf coordinates. By definition, the leaves are described by the identity $y = \text{const}$. Additionally, at $E_0$ we have $y = 0$. Taking into account the partition of coordinates, we shall assume that on the leaf $E_\alpha$ the one-forms spanning (5) are given as
\[ \omega_i|_{E_\alpha} = \sum_{j=1}^r d z_j. \]
Our objective lies in defining an augmenting kinematics map \( h(x) = (h_1(x), \ldots, h_r(x)) \), \( h|_{E_0} = id_{E_0} \), such that in a neighbourhood of the zero-leaf \( E_0 \), the collection of one-forms
\[
\bar{\omega}_i = \sum_{j=1}^r B_{ij}(x)dh_j(x)
\]
i = 1, \ldots, r satisfies the following conformity conditions
\[
\bar{\omega}_i|_{E_0} = \omega_i|_{E_0} \quad (10)
\]
and
\[
\omega_i(x)X(x) = \bar{\omega}_i(x)X(x). \quad (11)
\]

After setting
\[
\omega(x) = \begin{bmatrix}
\omega_1(x) \\
\vdots \\
\omega_r(x)
\end{bmatrix}, \quad \bar{\omega}(x) = \begin{bmatrix}
\bar{\omega}_1(x) \\
\vdots \\
\bar{\omega}_r(x)
\end{bmatrix}
\]
the condition (11) can be re-stated as
\[
\omega(x)X(x) = F(x) = B(x)\frac{\partial h(x)}{\partial x}X(x), \quad (12)
\]
where \( F_i(x) = \omega_i(x)X(x) \), and \( F(x) = F_1(x), \ldots, F_r(x) \). Let us observe that, since
\[
\bar{\omega}_i(x) = \sum_{j=1}^r B_{ij}(x)dh_j(x) = \sum_{j=1}^r \sum_{k=1}^m B_{ij}(x)\frac{\partial h_j(x)}{\partial y_k}dy_k + \sum_{j=1}^r \sum_{l=1}^r B_{ij}(x)\frac{\partial h_j(x)}{\partial z_l}dz_l,
\]
then
\[
\bar{\omega}_i|_{E_0} = \sum_{j=1}^r \sum_{l=1}^r B_{ij}(x)\frac{\partial h_j(x)}{\partial z_l}dz_l.
\]

Using the last identity, from the condition (10) and (9) we deduce that
\[
B(x)\frac{\partial h(x)}{\partial z} = I_r. \quad (13)
\]
Now, setting \( x = \Phi_s(y,z) \), and taking into account (8) and (12), we compute
\[
B(\Phi_s(y,z))\frac{\partial h \circ \Phi_s(y,z)}{\partial x}X(\Phi_s(y,z)) = B(\Phi_s(y,z))\frac{\partial h \circ \Phi_s(y,z)}{\partial x}d\Phi_s(y,z)\frac{ds}{ds},
\]
and conclude that
\[
sB(\Phi_s(y,z))\frac{dh \circ \Phi_s(y,z)}{ds} = F \circ \Phi_s(y,z). \quad (14)
\]
Obviously, the identity (13) yields

\[ B(\Phi_t(y,z)) \frac{\partial h \circ \Phi_t(y,z)}{\partial z} = I_r. \] (15)

To continue, we shall set \( H(s, y, z) = h \circ \Phi_s(y,z) \), where serves as a parameter. After a multiplication of both the identities (14) and (15) by the matrix \( B^{-1}(\Phi_s(y,z)) \), and a substitution for \( B^{-1}(\Phi_s(y,z)) \) from (15), we arrive at a system of identical partial differential equations parameterized by \( y \in \mathbb{R}^m \)

\[ \frac{\partial H_i}{\partial s} - \sum_{j=1}^r F_j(\Phi_s(y,z)) \frac{\partial H_i}{\partial z_j} = 0, \] (16)

defined for every \( i = 1, \ldots, r \). These equations can be solved by the method of characteristic. Indeed, it is easily seen that, if \( z(s) \) satisfies the differential equation

\[ \frac{dz}{ds} = -\frac{F(\Phi_s(y,z))}{s}, \quad z(0) = z_0, \] (17)

then the map \( H(s, y, z(s)) = \text{const} \). Using this observation and the properties of the homotopy map \( \Phi_t(x) \), we get

\[ H(1, y, z(1)) = h \circ \Phi_1(y,z) = h(y,z) = H(0, y, z(0)) = h \circ \Phi_0(y,z(0)) = h(0, z_0) = z_0, \]

so, in order to determine the augmenting kinematics map \( h(x) \) we need to solve the implicit equation

\[ z(1) = z = \varphi(1, y, h(y,z)), \] (18)

where \( \varphi(s, y, z_0) \) denotes the flow of (17), and to restore the original variables \( x \).

The defining procedure of the augmenting kinematics map \( h(x) \) can be summarized in the following way:

1. Given the kinematics with redundancy degree \( r \), and the Jacobian, find the codistribution (5).
2. Choose a foliation of the configuration space by \( r \)-dimensional leaves \( E_\alpha \) such that each generator of the codistribution (5) satisfies (9).
3. Establish a homotopy between the configuration space and the zero-leaf, and define the vector field (7).
4. Determine the map (12).
5. Formulate and solve the characteristic equation (17).
6. Compute the map \( h(x) \) from (18).

In the following sections we shall apply this procedure to the kinematics of a manipulator and of a mobile robot.

3. POLYCRANK

The POLYCRANK is a unique design of a very fast 7 degree-of-freedom manipulator, without joint limits, whose inertia matrix is close to diagonal. The manipulator, constructed at the Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology, was presented in [13] and [14]. A coordinate representation of the POLYCRANK’s kinematics, defined in the Cartesian position coordinates and the Euler angles ZXZ, is the following

\[
y = (y_1, y_2, \ldots, y_6) = k(x) = (l_1 c_1 + l_2 c_2 + l_6 c_3 + (l_4 s_4 + l_5 s_5) s_3, \\
l_1 s_1 + l_2 s_2 + l_6 s_3 - (l_4 s_4 + l_5 s_5) c_3, l_3 + l_4 c_4 + l_5 c_5, x_3, x_6, x_7),
\]

(19)

where \( l_i, i = 1, \ldots, 6 \) denote the lengths of the manipulator’s links, and \( s_i, c_i \) stand for \( \sin x_i \) and \( \cos x_i \). For the sake of simplicity we shall further assume that all \( l_i = 1 \). Under this assumption the manipulator’s Jacobian

\[
J(x) = \begin{bmatrix}
-s_1 & -s_2 & -s_3 + (s_4 + s_5) c_3 & s_3 c_4 & s_3 c_5 & 0 & 0 \\
c_1 & c_2 & c_3 + (s_4 + s_5) s_3 & -c_3 c_4 & -c_3 c_5 & 0 & 0 \\
0 & 0 & 0 & -s_4 & -s_5 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.
\]

The distribution associated with the Jacobian pseudo-inverse is spanned by the columns of the Jacobian transpose \( J^T(x) \). The codistribution that annihilates this distribution is generated by the single one-form

\[
\omega(x) = \left( -\frac{s_4 s_3}{s_5 s_2}, \frac{s_4 s_3}{s_5 s_2}, 0, 1, -\frac{s_4}{s_5}, 0, 0 \right),
\]

(20)
where $s_{ij} = \sin(x_i - x_j)$. The form $\omega(x)$ is well defined outside jointspace singularities $x_5 = j\pi$ or $x_2 - x_1 = l\pi$, $j, l = 0, \pm 1, \ldots$. In a region of the jointspace free from singularities we define a foliation with 1-dimensional leaves

$$E_{a,b,c,a} = \{a_1 + a\} \times \{a_2 + b\} \times \{a_3\} \times R \times \{a_4\} \times \{a_5 + c\} \times \{a_6\} \times \{a_7\},$$

where $a, b, c \neq 0, a \neq b$ and $a_i \in R$, so that the zero-leaf $E_{a,b,c,0} = \{a\} \times \{b\} \times \{0\} \times R \times \{c\} \times \{0\} \times \{0\}$. As in the previous case, parameters $a, b, c$ should keep the zero-leaf away from singular configurations. The homotopy can be chosen as $\Phi_t(x) = tx + (1 - t)(a, b, 0, x_4, c, 0)^T = (t(x_1 - a) + a, t(x_2 - b) + b, tx_3, x_4, t(x_5 - c) + c, tx_6, tx_7)^T$.

On the basis of this homotopy we compute the vector field

$$X(x) = \frac{d\Phi_t(x)}{dt}\big|_{t=1} = (x_1 - a, x_2 - b, x_3, 0, x_5 - c, x_6, x_7)^T$$

that, acted on by the one-form (20), results in the function

$$\omega(x)X(x) = F(x) = -\frac{sx_4s_32(x_1 - a) + s_4s_31(x_2 - b) - s_4s_21(x_5 - c)}{s_5s_21}.$$

Now we re-define the variables in such a way that $z = x_4$ represents the leaf coordinate, and $y_i = x_i, i = 1, 2, 3$ and $y_{i+3} = x_{i+4}, i = 1, 2, 3$ refer to the coordinates transverse to the leaf, and obtain the characteristic equation (24) in the following form

$$\frac{dz}{ds} = \frac{(y_1 - a)\sin(s(y_4 - c) + c - z)\sin(s(y_3 - s(y_2 - b) - b) + (y_4 - c)\sin z}{\sin(s(y_4 - c) + c)\sin(s(y_2 - b) - b - s(y_1 - a) - a} + \frac{(y_2 - b)\sin(s(y_4 - c) + c - z)\sin(s(y_3 - s(y_1 - a) - a) - (y_2 - b)\sin(s(y_4 - c) + c)\sin(s(y_2 - b) - b - s(y_1 - a) - a).$$

(21)

Let $\varphi(s, y, z_0)$ denote the flow of (21). Then, the augmenting kinematics map $h(x) = h(y, z)$ defining the extended Jacobian inverse is obtained by solving numerically the implicit equation

$$z = \varphi(1, y, h(y, z)).$$

Results of computer simulations of the extended Jacobian and of the Jacobian pseudo-inverse inverse kinematics algorithms for POLYCRANK will be presented in the next section.
4. Chained-form system

We shall consider the chained-form system

\[ \begin{align*} 
\dot{x}_1 &= u_1, \\
\dot{x}_2 &= u_2, \\
\dot{x}_3 &= x_2 u_1, 
\end{align*} \]  

(22)

representing a feedback equivalent of the differential drive-type mobile platform. Let \( x(t) = \phi_{x_0} (u(t)) \) denote the state trajectory of this system steered by a control function \( u(t) \). Then, given a control time horizon \( T > 0 \), the kinematics are identified with the end point map of the system (22), defined as \( x(T) = \phi_{x_0} (u(t)) \). Because the domain of the kinematics is an infinite dimensional functional space, for computational purposes we need to use a finite dimensional expansion of control functions. In this way the kinematics can be treated as a map between finite dimensional spaces. For the chained-form system we have adopted a 5-dimensional control space in the form

\[ u_1(t) = \lambda_1 + \lambda_2 \sin \frac{2\pi t}{T}, \quad u_2(t) = \lambda_3 + \lambda_4 \sin \frac{2\pi t}{T} + \lambda_5 \cos \frac{2\pi t}{T}. \]

An insertion of this control into (22) and a substitution \( x_0 = 0 \), \( T = 2\pi \) results in the kinematics

\[ x(T) = K(\lambda) = \left( T\lambda_1, T\lambda_3, \frac{1}{2} T^2 \lambda_1 \lambda_3 - T \lambda_2 \lambda_3 + T \lambda_1 \lambda_4 + \frac{1}{2} T \lambda_2 \lambda_5 \right)^T, \]

and the Jacobian

\[ J(\lambda) = T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{1}{2} T \lambda_3 + \lambda_4 & -\lambda_3 + \frac{1}{2} \lambda_5 & \frac{1}{2} T \lambda_1 - \lambda_2 & \lambda_1 & \frac{1}{2} \lambda_2 \end{bmatrix}. \]

The distribution associated with the Jacobian pseudo-inverse is generated by the columns of the Jacobian transpose divided by \( T \),

\[ J^T(\lambda) = \begin{bmatrix} 1 & 0 & \frac{1}{2} T \lambda_3 + \lambda_4 \\ 0 & 0 & -\lambda_3 + \frac{1}{2} \lambda_5 \\ 0 & 1 & \frac{1}{2} T \lambda_1 - \lambda_2 \\ 0 & 0 & \lambda_1 \\ 0 & 0 & \frac{1}{2} \lambda_2 \end{bmatrix}. \]
Under assumption that $\lambda_1 > 0$ we find two one-forms annihilating this distribution
\[ \omega_1(\lambda) = \left(0, 1, 0, \frac{\lambda_1 - \lambda_3}{\lambda_1}, 0\right), \quad \omega_2(\lambda) = \left(0, 0, 0, -\frac{\lambda_1}{\lambda_3}, 1\right). \quad (23) \]

Next, we introduce a foliation of the control space by 2-dimensional leaves of the form
\[ E_{\alpha_1, \alpha_2} = \{\alpha_1 + a\} \times R \times \{\alpha_2\} \times \{\alpha_3\} \times R, \]
where $a > 0$, and for $i = 1, 2, 3$, $\alpha_i \in R$, with zero-leaf $E_{\alpha_1, 0} = \{a\} \times R \times \{0\} \times \{0\} \times R$.

The role of the parameter $a$ consists in placing the zero-leaf reasonably far from the singularity $\lambda_1 = 0$. Having defined the zero-leaf, we choose the homotopy $\Phi_t(\lambda) = t\lambda + (1-t)(a, \lambda_2, 0, 0, \lambda_3)^T = (t\lambda_1 + (1-t)a, \lambda_2, t\lambda_3, t\lambda_4, \lambda_5)$, compute the vector field
\[ X(\lambda) = \frac{d\Phi_t(\lambda)}{dt} \big|_{t=1} = (\lambda_1 - a, 0, \lambda_3, \lambda_4, 0)^T, \]
and obtain a pair of functions
\[ (\omega_1, X)(\lambda) = F_1(\lambda) = \frac{\lambda_1}{\lambda_3} (\lambda_3 - \frac{a}{\lambda_3}) \quad \text{and} \quad (\omega_2, X)(\lambda) = F_2(\lambda) = -\frac{\lambda_1}{\lambda_3}.\]

Observe that a similar procedure can be adopted in the case of $\lambda_1 < 0$. To proceed further, let us re-define variables in such a way that $z_1, z_2$ denote the coordinates along the leaves, so we set $y_1 = \lambda_1 - a, z_1 = \lambda_2, y_2 = \lambda_3, y_3 = \lambda_4$, and $z_5 = \lambda_5$. Using the fact that in new variables the homotopy map
\[ \Phi_s(y, z) = (sy_1 + (1-s)a, z_1, sy_2, sy_3, z_2), \]
we derive the characteristic equation
\[ \frac{dz}{ds} = \frac{y_3}{2(y_1 + a)} \begin{bmatrix} 0 & \frac{y_1}{y_1 + a} & \frac{y_3}{y_1 + a} \\ 1 & 0 & 0 \end{bmatrix} z - \frac{y_3 z_3}{y_1 + a}. \quad (24) \]

This is a linear differential equation with respect to the leaf coordinates $z$, parameterized by $y$. After some analysis the following close form solution to (24) has been
discovered
\[ z_1(s) = \frac{1}{2} \left( (sy_1 + a) \sinh \frac{3\pi}{4} + \frac{3\pi}{4} + (sy_1 + a) \cosh \frac{3\pi}{4} \right) z_{10} + \]
\[ \frac{1}{2} \left( (sy_1 + a) \frac{3\pi}{4} a - \frac{3\pi}{4} - (sy_1 + a) \frac{3\pi}{4} a \right) \right) z_{20} - \]
\[ \frac{4e\pi y_3}{4y_1 - y_3^2} (sy_1 + a) + \frac{2e\pi}{2y_1 + y_3} (sy_1 + a) \frac{3\pi}{4} - \frac{2e\pi}{2y_1 + y_3} (sy_1 + a) \frac{3\pi}{4}, \]
\[ z_2(s) = \frac{1}{2} \left( (sy_1 + a) \frac{3\pi}{4} a - \frac{3\pi}{4} - (sy_1 + a) \frac{3\pi}{4} a \right) z_{10} + \]
\[ \frac{1}{2} \left( (sy_1 + a) \frac{3\pi}{4} a - \frac{3\pi}{4} + (sy_1 + a) \frac{3\pi}{4} a \right) \right) z_{20} + \]
\[ \frac{8e\pi y_3 - 2e\pi y_3^2}{4y_1 - y_3^2} + \frac{2e\pi}{2y_1 + y_3} (sy_1 + a) \frac{3\pi}{4} + \frac{2e\pi}{2y_1 + y_3} (sy_1 + a) \frac{3\pi}{4}. \]
\[ (25) \]

In accordance with (18), this solution will produce a pair of functions
\[ h_1(\lambda) = \left( \lambda_2 + \frac{4\lambda_3 \lambda_4}{4(\lambda_1 - a)^2 - \lambda_2^2 - \lambda_4^2} \right) \cosh \left( \frac{\lambda_1}{2(\lambda_1 - a)^2} \ln \left( \frac{\lambda_4}{\lambda_1} \right) - \right) \]
\[ \left( \lambda_5 + \frac{8a\lambda_1 \lambda_2}{4(\lambda_1 - a)^2 - \lambda_2^2 - \lambda_4^2} \right) \sinh \left( \frac{\lambda_1}{2(\lambda_1 - a)^2} \ln \left( \frac{\lambda_4}{\lambda_1} \right) \right) \right) - \frac{4\lambda_3 \lambda_4}{4(\lambda_1 - a)^2 - \lambda_2^2}, \]
\[ (26) \]
\[ h_2(\lambda) = - \left( \lambda_2 + \frac{4\lambda_3 \lambda_4}{4(\lambda_1 - a)^2 - \lambda_2^2 - \lambda_4^2} \right) \sinh \left( \frac{\lambda_1}{2(\lambda_1 - a)^2} \ln \left( \frac{\lambda_4}{\lambda_1} \right) \right) + \]
\[ \left( \lambda_5 + \frac{8a\lambda_1 \lambda_2}{4(\lambda_1 - a)^2 - \lambda_2^2 - \lambda_4^2} \right) \cosh \left( \frac{\lambda_1}{2(\lambda_1 - a)^2} \ln \left( \frac{\lambda_4}{\lambda_1} \right) \right) \right) - \frac{8a\lambda_3 \lambda_1}{4(\lambda_1 - a)^2 - \lambda_2^2}. \]

that determine the augmenting kinematics map \( h(\lambda) = h_1(\lambda), h_2(\lambda) \), and define the extended Jacobian inverse. The extended Jacobian inverse kinematics algorithm assumes the form of the dynamic system
\[ \frac{d\lambda}{d\theta} = -J^{\text{EE}}(\lambda) (K(\lambda) - y_d), \]
\[ (27) \]
where the inverse \( J^{\text{EE}}(\lambda) \) is computed from (4), after a substitution from (3) and (26). In the next section we shall illustrate the performance of this algorithm with computer simulations.

5. Computer simulations

5.1. POLYCRANK

Computer simulations have been accomplished for the following initial data: \( x_0 = \left( \frac{\pi}{6}, \frac{\pi}{6}, \frac{\pi}{6}, \frac{\pi}{6}, \frac{\pi}{6} \right) \), \( y_0 = (1, 2, 1, 1.5708, 1.0472, 0.5236) \), \( a = \frac{\pi}{2}, b = \pi, c = \frac{\pi}{2}, \gamma = 0.3 \). The position and orientation trajectories generated by \( J^{\text{EE}} \) and \( J^{\text{EP}} \) are shown in
figures 1, 2, and in figures 3 and 4. The convergence of configuration variables has been visualized in figures 5, 6 as well as in figures 7 and 8. It can be seen that, although the convergence of the extended Jacobian algorithm is slower than that of the Jacobian pseudo-inverse, the resulting taskspace trajectories resemble each other quite well.

![Figure 1: $J^{FE}$: position trajectory of POLYCRANK](image)

5.2. Chained-form system

Computer simulations of the extended Jacobian inverse kinematics algorithm (27) have been accomplished for the initial state $x_0 = (0, 0, 0)$, and a pair of desirable terminal states $y_{d1} = (5, -5, -2)$ and $y_{d2} = (5, 5, -2)$ of the chained form system. The initial values of control parameters have been picked up as $\lambda_0 = (0.5, 0.2, 0.5, 0.2, 0.2)$, the control time $T = 2\pi$, the convergence coefficient $\gamma = 0.3$. We have set $a = 100$. The performance of the extended Jacobian algorithm has been compared with that of the Jacobian pseudo-inverse algorithm. The results are demonstrated in the following figures. Figures 9, 10, and figures 11, 12 show the configuration space trajectories.
Figure 2: $J^P$: position trajectory of POLYCRANK

Figure 3: $J^E$: orientation trajectory of POLYCRANK
Figure 4: $J^P$: orientation trajectory of POLYCRANK

Figure 5: $J^E$: convergence of $x_1 - x_4$ for POLY

Figure 6: $J^P$: convergence of $x_1 - x_4$ for POLY-CRANK

Figure 7: $J^E$: convergence of $x_1 - x_4$ for POLY-CRANK
generated by the extended Jacobian and the Jacobian pseudo-inverse algorithms, respectively, for $y_{d1}$ and $y_{d2}$. The corresponding convergence of solutions to the inverse kinematics problem are displayed in the figures 13-16 as well as in the figures 17-20. These examples demonstrate that, depending on the location of the configuration space trajectory with respect to a manifold on which the codistributions associated with $J^P$ and $J^E$ coincide, the performance of these two algorithms is either very similar (the case of $y_{d1}$) or quite different (the case of $y_{d2}$). Despite this, qualitatively, the configuration space trajectories produced by $J^P$ and $J^E$ resemble each other in both the cases under consideration. The extended Jacobian algorithm is repeatable, and, at the same time, its speed of convergence is comparable with that of the Jacobian pseudo-inverse, so the objectives of the approximation have been achieved.

6. Conclusions

We have proposed a method of synthesis of extended Jacobian inverse kinematics algorithms, based on the approximation of the Jacobian pseudo-inverse. The mathematical machinery supporting this method comes from the differential geometric approximation of codistributions. The approximation guarantees that the approximant and the approximated codistribution coincide on leaves of a certain foliation of the configuration space, and along a characteristic vector field. These requirements translate into a system of linear partial differential equations whose characteristic curves provide
Figure 9: $J_{EE}$: configuration space trajectory for $y_d$.

Figure 10: $J_{EP}$: configuration space trajectory for $y_d$. 
Figure 11: $J_E^E$: configuration space trajectory for $y_{d2}$

Figure 12: $J_P^P$: configuration space trajectory for $y_{d2}$
Figure 13: $J^E_1$: convergence of $\lambda_1, \lambda_2$ for $y_{d1}$

Figure 14: $J^P_1$: convergence of $\lambda_1, \lambda_2$ for $y_{d1}$

Figure 15: $J^E_2$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d1}$

Figure 16: $J^P_2$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d1}$

Figure 17: $J^E_2$: convergence of $\lambda_1, \lambda_2$ for $y_{d2}$

Figure 18: $J^P_2$: convergence of $\lambda_1, \lambda_2$ for $y_{d2}$
the augmenting kinematics map. The method has been illustrated with the derivation of the extended Jacobian inverse kinematics algorithm for the POLYCRANK stationary robot and for a chained-form mobile robot. Computer simulations demonstrate the performance and the convergence of the extended Jacobian inverse kinematics algorithm, in comparison with the Jacobian pseudo-inverse algorithm. They foster a conclusion that the objectives of the approximation have been achieved: the constructed inverse kinematics algorithm is repeatable, unlike the Jacobian pseudo-inverse, and quickly convergent, like the Jacobian pseudo-inverse.

7. Acknowledgments

This research was supported by the Foundation for Polish Science.

References


21
Figures:

Figure 1: $J^{RE}$: position trajectory of POLYCRANK
Figure 2: $J^P$: position trajectory of POLYCRANK

Figure 3: $J^E$: orientation trajectory of POLYCRANK
Figure 4: $J^{yP}$: orientation trajectory of POLYCRANK

Figure 5: $J^{yE}$: convergence of $x_1 - x_4$ for POLY

Figure 6: $J^{yP}$: convergence of $x_1 - x_4$ for POLY-CRANK

25
Figure 7: $J^E$: convergence of $x_5 - x_6$ for POLY-CRANK

Figure 8: $J^P$: convergence of $x_5 - x_6$ for POLY-CRANK

Figure 9: $J^E$: configuration space trajectory for $y_{d1}$
Figure 10: $J^p$: configuration space trajectory for $y_{d1}$

Figure 11: $J^b$: configuration space trajectory for $y_{d2}$
Figure 12: $J^P$: configuration space trajectory for $y_{d2}$

Figure 13: $J^E$: convergence of $\lambda_1$, $\lambda_2$ for $y_{d1}$

Figure 14: $J^P$: convergence of $\lambda_1$, $\lambda_2$ for $y_{d1}$
Figure 15: $J^E$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d1}$

Figure 16: $J^P$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d1}$

Figure 17: $J^E$: convergence of $\lambda_1, \lambda_2$ for $y_{d2}$

Figure 18: $J^P$: convergence of $\lambda_1, \lambda_2$ for $y_{d2}$

Figure 19: $J^E$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d2}$

Figure 20: $J^P$: convergence of $\lambda_3, \lambda_4, \lambda_5$ for $y_{d2}$