A Modified DLS Scheme with Controlled Cyclic Solution for Inverse Kinematics in Redundant Robots

Mohammad Safeea, Richard Bearee, and Pedro Neto

***Abstract*— Redundancy in robotic manipulators has many advantages, it is successfully used to achieve better dexterity, to avoid obstacles, singularities or the kinematic limitations. However, redundancy makes the Inverse Kinematics (IK) problem harder to solve. The Damped Least Squares (DLS) is a powerful method for calculating the inverse kinematics of redundant robots, but it suffers from non-cyclicity issue, where, a closed curve motion in the Cartesian space of the end-effector (EEF) does not map into a closed curve in the joints space. This results in non repetitive motion in the joints space even though the EEF motion is repetitive. In this study we present a solution for the non-cyclicity problem in DLS method. The proposed scheme was tested successfully both in simulation (9 DoF robot) and on a real KUKA iiwa (7 DoF robot).**

Keywords: Redundant manipulators, cyclicity, damped least squares, inverse kinematics

I. Introduction

Redundancy is a term used for referring to robots that have more degrees of freedom (n) than the minimal degrees of freedom (m) necessary to complete a given task [1]. The extra degrees of freedom $(n - m)$ offer better flexibility and can be used to achieve multiple objectives, for example to avoid collisions with obstacles nearby [2, 3, 4]. To avoid singularities and joint limits [5, 6]. Also, for improving the dynamical response [7], for minimizing energy consumption [8] and for improving dexterity [9], or for performing secondary (lower priority) tasks [10]. In robot control, the Inverse Kinematics (IK) problem is defined as calculating the joints angles that correspond to certain EEF path in the Cartesian space. For redundant manipulators, various methods have been proposed for calculating the inverse kinematics, the majority are based on instantaneous or local resolution of redundancy at the velocity level by utilising the Jacobian matrix [11]. The pseudo inverse method [12] is successfully used for this purpose. However, the pseudo inverse method, without further modification, suffers from the singularity problem, as mentioned in [13]. Consequently, researchers utilised modified pseudo inverses such that a motion in the null space of the redundant manipulator is used to optimize a dexterity measure [14] which results in avoiding singular configurations. Other researchers used a damping factor as in the Damped Least Squares (DLS) method [15].

Nevertheless, the pseudo inverse based methods (including DLS) suffer from the non-cyclicity problem [16, 17]. Where, a closed curve motion in the Cartesian space of the end-effector (EEF) does not map into a closed curve in the joints space of the redundant robot. This causes a deviation in the robot configuration with time. Consequently, a repetitive motion in the Cartesian space does not result in a repetitive motion in the joints space, this drawback becomes more critical as the number of redundant DoF increases.

The extended Jacobian method presented in [13] offers a solution to the non-cyclic behaviour of the inverse kinematic in redundant manipulators, in this method the Jacobian of the robot is extended from an $(m \times n)$ matrix to an $(n \times n)$ by adding extra $(n–m)$ linear equations into the IK problem for optimising additional criteria. However, this method suffers from two major disadvantages which are mentioned in [18] section "6.5 Inverse kinematic model of redundant robots":

1) The choice of the $(n–m)$ additional relationships is not a trivial matter;
2) The extended Jacobian may be singular even though the end-effector Jacobian is of full rank. These configurations are called artificial singularities or algorithmic singularities.

Recently, various researchers investigated the use of neural networks for achieving a repetitive motion planning [19, 20, 21]. However, the pseudo inverse based methods have various advantages over the neural network methods including more accurate solution, lower computational cost, simultaneous control of EEF position and orientation, in addition to its applicability even for very high DoF. The pseudo inverse based methods proved to be quite robust so they are used widely in robot simulators and for controlling industrial manipulators.

In this article we propose a novel damped least squares scheme for achieving controlled cyclic behaviour of the inverse kinematics problem in redundant manipulators, while at the same time preserving immunity to singularities. Rather than treating the problem at the joints velocity level, the problem is treated at the joints position level, very accurate solution of joints angles is achieved efficiently, which is very important for robot simulation and for the control of industrial manipulators.

II. Motivation and contribution

The Damped Least Squares [15, 22] is a powerful pseudo inverse based method which is used to calculate the inverse

Mohammad Safeea is with the University of Coimbra, Portugal, and with Arts et Métiers, ParisTech, France, e-mail: ms@uc.pt.

Richard Bearee is with Arts et Métiers, ParisTech, Lille, France, e-mail: Richard.BEAREE@ENSAM.EU.

Pedro Neto is with the Department of Mechanical Engineering, University of Coimbra, Coimbra, Portugal, e-mail: pedro.neto@dem.uc.pt.

kinematics of redundant robots. It is successfully applied for precise tracking of EEF Cartesian motion. It does not suffer from singularity problem. Also, this method is more efficient computationally than the modified pseudo inverses, because it does not require calculating a manipulability index and its gradient. Owing to its various advantages, the DLS method is widely used in many applications including the well known robotics simulator Virtual Robot Experimentation Platform (V-REP) [23]. By implementing DLS in an iterative scheme [24], the method can be used to calculate joints angles for a particular EEF pose quite accurately. Though, the problem of the non-cyclic behaviour at the joints space in DLS method still persists. Figure 1 shows a simulation using Vrep, where Vrep's own DLS solver is used to calculate the inverse kinematics of a redundant robot (9 DoF) moving in a repetitive circular motion at the EEF. Simulation video is available in the multimedia material. After few cycles it is noticed that the configuration of the robot deviates resulting in non repetitive motion of the joints. Figure 2 shows the non-cyclic behaviour in the joint angles for this simulation.

In this study, we address the non-cylicity problem of DLS method, consequently, our proposed scheme offers various advantages over existing methods including:

1) The proposed scheme produces controlled cyclic motion, with predictable configurations of the redundant robot;
2) The proposed scheme is immune to singularities, virtue to DLS method which is the backbone of our proposed algorithm;
3) The proposed scheme acts at the joints position level, it provides accurate joint angle solution, resulting in accurate tracking of EEF path, which is very important for robot simulators or for the control of real robots (for example the computed torque control method [25] which requires IK resolution for calculating the reference joint angles from the EEF path).

III. Mathematical formulation

The differential kinematics equation of the robotic manipulator [26],[27] is:

$$\dot{\boldsymbol{x}} = \mathbf{J}\dot{\boldsymbol{q}} \tag{1}$$

Where $\dot{\boldsymbol{q}}$ is the joints velocities vector, $\dot{\boldsymbol{x}}$ is the end-effector's velocity and $\mathbf{J}$ is the manipulator's Jacobian matrix, each element of which:

$$J_{ij} = \frac{\delta f_i}{\delta q_j} \tag{2}$$

Where $\boldsymbol{f}$ is the forward kinematics function of the robotic manipulator. Multiplying both sides of (1) by $\triangle t$ yields:

$$\triangle \boldsymbol{x} \approx \mathbf{J} \triangle \boldsymbol{q} \tag{3}$$

Which is the backbone for the iterative methods used to solve the IK problem [24].

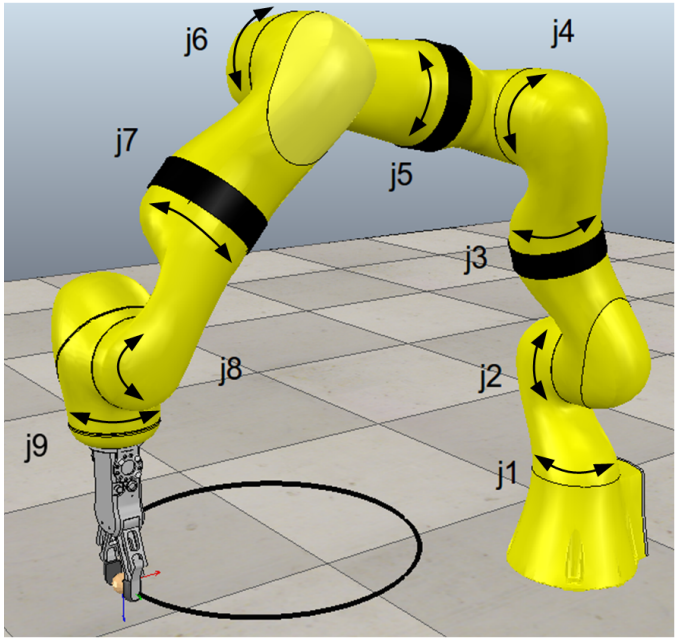


Fig. 1
Redundant manipulator with nine joints (9 DoF) used for simulations.

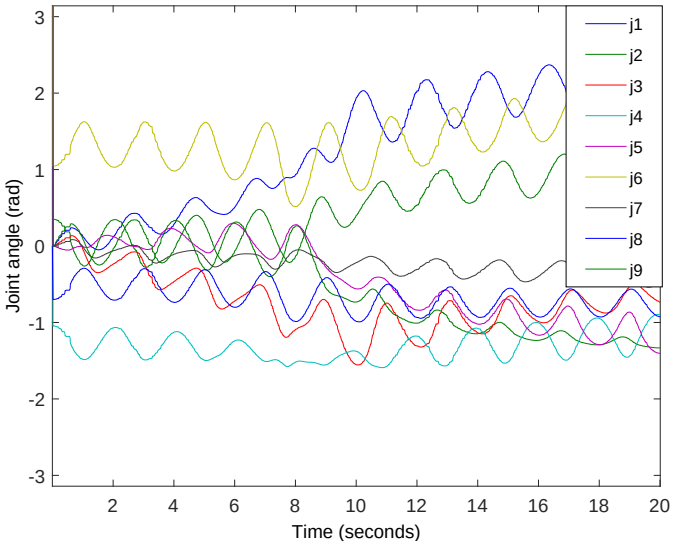


Fig. 2
Joint angles with time, the inverse kinematics module (DLS solver) of Vrep is used for tracking the EEF position.

Algorithm 1 Iterative Damped Least Squares algorithm.

Input : $\boldsymbol{q}_{ini}$ joints positions initial guess
$\boldsymbol{x}_g$ goal pose of EEF
Output : $\boldsymbol{q}_g$ joitns angles solution

01 : $\boldsymbol{x}_{ini} := \boldsymbol{f}(\boldsymbol{q}_{ini})$ *% Forward kinematics*
02 : $\boldsymbol{q} := \boldsymbol{q}_{ini}$
03 : $\boldsymbol{e} := (\boldsymbol{x}_{ini} - \boldsymbol{x}_g)$ *% Calculate error*
04 : **while** $(norm(\boldsymbol{e}) > \boldsymbol{\varepsilon})$ *% While the error is big*
05 : $\boldsymbol{q} := \boldsymbol{q} - \mathbf{J}^*\boldsymbol{e}$ *% Angles update equation*
06 : $\boldsymbol{x} := \boldsymbol{f}(\boldsymbol{q})$ *% Forward kinematics*
07 : $\boldsymbol{e} := (\boldsymbol{x} - \boldsymbol{x}_g)$ *% Calculate error*
08 : **Loop**
09 : $\boldsymbol{q}_g := \boldsymbol{q}$

Algorithm 2 The proposed scheme.

Input : $\boldsymbol{q}_f$ joints positions feedback at current instnant
$\boldsymbol{q}_s$ joints positions at starting configuration
$\boldsymbol{x}_g$ goal pose of EEF
Output : $\boldsymbol{q}_g$ joitns angles solution

01 : $\boldsymbol{x}_f := \boldsymbol{f}(\boldsymbol{q}_f)$ *% Forward kinematics*
02 : $\boldsymbol{q} := \boldsymbol{q}_f$
03 : $\boldsymbol{e} := (\boldsymbol{x}_f - \boldsymbol{x}_g)$ *% Calculate error*
04 : **while** $(norm(\boldsymbol{e}) > \boldsymbol{\varepsilon})$ *% While the error is big*
05 : $\boldsymbol{q} := \boldsymbol{q}_s + (\mathbf{I} - \kappa\mathbf{N})(\boldsymbol{q} - \boldsymbol{q}_s) - \mathbf{J}^*\boldsymbol{e}$ *% Angles update equation*
06 : $\boldsymbol{x} := \boldsymbol{f}(\boldsymbol{q})$ *% Forward kinematics*
07 : $\boldsymbol{e} := (\boldsymbol{x} - \boldsymbol{x}_g)$ *% Calculate error*
08 : **Loop**
09 : $\boldsymbol{q}_g := \boldsymbol{q}$

A. Iterative Damped Least Squares

Applying Levenberg-Marquardt [28, 29] on (3) yields the damped least squares, consequently, an approximation of $\triangle\boldsymbol{q}$ for a change $\triangle\boldsymbol{x}$ can be calculated:

$$\triangle\boldsymbol{q} \approx \mathbf{J}^*\triangle\boldsymbol{x} \tag{4}$$

Where $\mathbf{J}^* = \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda^2\mathbf{I})^{-1}$, λ is a damping factor, $\mathbf{I}$ is the identity matrix. It is noticed that the term $(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \lambda^2\mathbf{I})$ is always invertible. Consequently, the DLS method does not suffer from singularity issues. Based on (4), an iterative method can be used for calculating the joint angles at a goal pose $\boldsymbol{x}_g$. The joints update assignment is given by:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \mathbf{J}^*(\boldsymbol{x}_k - \boldsymbol{x}_g) \tag{5}$$

Where $\boldsymbol{q}_k, \boldsymbol{x}_k$ are the joints angles and the EEF pose (respectively) at iteration k. This iterative method requires providing an initial guess $\boldsymbol{q}_{ini}$. We describe this iterative method in Algorithm 1.

B. Proposed Scheme

The proposed scheme is built by modifying Algorithm 1, mainly by using a different joints angles update equation:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_s + (\mathbf{I} - \kappa\mathbf{N})(\boldsymbol{q}_k - \boldsymbol{q}_s) - \mathbf{J}^*(\boldsymbol{x}_k - \boldsymbol{x}_g) \tag{6}$$

Where $\boldsymbol{q}_s$ is the joints angles at the starting configuration at which the robot begins moving on the Cartesian path, $\mathbf{N}$ is the null space projection matrix and κ is a positive constant (its permissible range of values is shown in the section III-C - Stability analysis). Thus, using the vector $\boldsymbol{q}_s$ and the joints angular position feedback $\boldsymbol{q}_f$ a controlled cyclic motion of the robot can be calculated by implementing our scheme (illustrated in Algorithm 2). It is noticed that the computational cost (memory/execution time) of the proposed scheme is equal to the cost of the DLS method in addition to an extra cost mainly due to the term associated with the null space projection, line 05 of the Algorithm 2. This extra cost can be minimised when the singular value decomposition of the Jacobian matrix is used for calculating $\mathbf{N}$ and $\mathbf{J}^*$.

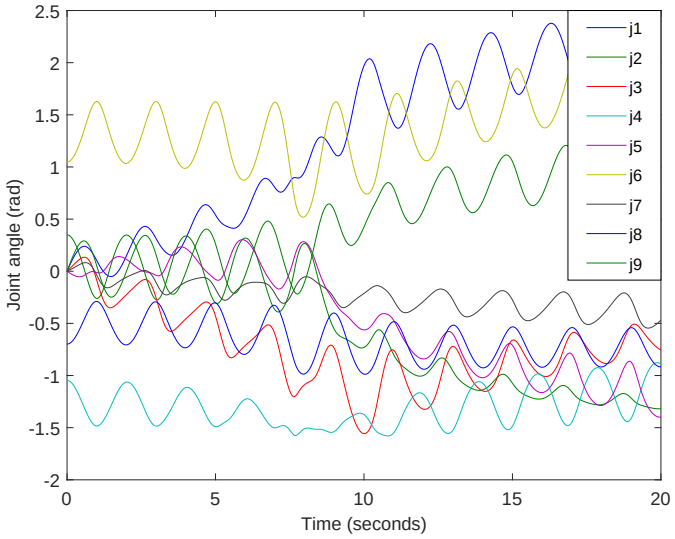


Fig. 3
TEST 1: JOINT ANGLES WITH TIME FOR LEGACY DLS METHOD.

C. Stability analysis

In this section, a proof for the stability of the proposed scheme is provided. However, first we show useful formulations for calculating the null space projection matrix, which will be important later in this section for proving the stability.

1) Null space projection matrix: For a redundant manipulator, $n > m$, the Jacobian matrix has more columns than rows, consequently it possess a null space projection matrix given by:

$$\mathbf{N} = \mathrm{I} - \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}})^{-1}\mathbf{J} \tag{7}$$

This formula suffers from a problem when the matrix $(\mathbf{J}\mathbf{J}^{\mathrm{T}})$ is singular. However, using the singular value decomposition on the Jacobian matrix:

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{8}$$

Where $\mathbf{S}$ is a diagonal matrix of the singular values of the Jacobian, $\mathbf{U}$ is an $m \times m$ orthonormal matrix (its column vectors are of unit length and normal to each other), and $\mathbf{V}$ is an $n \times m$ orthonormal matrix. By substituting (8) in (7) and noticing that $(\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathrm{I})$ and $(\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathrm{I})$ we find that:

$$\mathbf{N} = \mathrm{I} - \mathbf{V}\mathbf{V}^{\mathrm{T}} \tag{9}$$

This formula is used in our proposed scheme for calculating $\mathbf{N}$. On the other hand, we can extend $\mathbf{V}$ into an $n \times n$ orthonormal matrix $\mathbf{\Gamma}$, thus:

$$\mathbf{N} = \mathbf{I} - \mathbf{\Gamma}\mathbf{L}\mathbf{\Gamma}^{\mathrm{T}} \tag{10}$$

Where $\mathbf{L}$ is a diagonal matrix, with the first m diagonal elements equal to one, and the rest diagonal elements are equal to zero, $\mathbf{\Gamma}$ is an $n \times n$ orthonormal matrix with its the first m columns equal to the columns of $\mathbf{V}$. This formula is important for the stability proof.

2) Stability analysis: In this subsection the stability of the proposed method is proven. By rearranging the joint angles update equation (6) of the proposed method, we have:

$$\boldsymbol{q}_{k+1} - \boldsymbol{q}_s = (\mathbf{I} - \kappa\mathbf{N})(\boldsymbol{q}_k - \boldsymbol{q}_s) - \mathbf{J}^{*}(\boldsymbol{x}_k - \boldsymbol{x}_g) \tag{11}$$

From deferential kinematics we can write:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \mathbf{J}\left(\boldsymbol{q}_{k+1} - \boldsymbol{q}_k\right) \tag{12}$$

By subtracting $\boldsymbol{x}_g$ from both sides:

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_g = \boldsymbol{x}_k - \boldsymbol{x}_g + \mathbf{J}\left(\boldsymbol{q}_{k+1} - \boldsymbol{q}_k\right) \tag{13}$$

We substitute (11) in (13) and by noticing that $\mathbf{J}\mathbf{N} = 0$ then:

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_g = \boldsymbol{x}_k - \boldsymbol{x}_g - \mathbf{J}\mathbf{J}^{\bigstar}(\boldsymbol{x}_k - \boldsymbol{x}_g) \tag{14}$$

Or:

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_g = \left(\mathbf{I} - \mathbf{J}\mathbf{J}^{\bigstar}\right)(\boldsymbol{x}_k - \boldsymbol{x}_g) \tag{15}$$

By combining equations (15) and (11) together we have:

$$\begin{bmatrix} \boldsymbol{x}_{k+1} - \boldsymbol{x}_g \\ \boldsymbol{q}_{k+1} - \boldsymbol{q}_s \end{bmatrix} = \begin{bmatrix} \left(\mathbf{I} - \mathbf{J}\mathbf{J}^{\bigstar}\right) & 0 \\ -\mathbf{J}^{\bigstar} & (\mathbf{I} - \kappa\mathbf{N}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k - \boldsymbol{x}_g \\ \boldsymbol{q}_k - \boldsymbol{q}_s \end{bmatrix} \tag{16}$$

If we denote $\boldsymbol{u}_k = \begin{bmatrix} (\boldsymbol{x}_k - \boldsymbol{x}_g)^{\mathrm{T}} & (\boldsymbol{q}_k - \boldsymbol{q}_s)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ then:

$$\boldsymbol{u}_{k+1} = \begin{bmatrix} \left(\mathbf{I} - \mathbf{J}\mathbf{J}^{\bigstar}\right) & 0 \\ -\mathbf{J}^{\bigstar} & (\mathbf{I} - \kappa\mathbf{N}) \end{bmatrix} \boldsymbol{u}_k \tag{17}$$

Or:

$$\boldsymbol{u}_{k+1} = \mathbf{A}\boldsymbol{u}_k \tag{18}$$

The previous dynamical system is stable when the eigenvalues of the matrix $\mathbf{A}$ are anywhere from zero to one. In the following we prove that if the constant κ satisfies $0 \leq \kappa \leq 1$ the system is always stable. If we denoted A_{11} to the upper left block of matrix $\mathbf{A}$, then for calculating $\mathrm{A}_{11} = \mathbf{I} - \mathbf{J}\mathbf{J}^{\bigstar}$ we use the SVD decomposition of the Jacobian matrix, thus we find that:

$$\mathrm{A}_{11} = \mathbf{I} - \mathbf{U}\mathbf{S}^2\left(\mathbf{S}^2 + \lambda^2\mathbf{I}\right)^{-1}\mathbf{U}^{\mathrm{T}} \tag{19}$$

Or:

$$\mathrm{A}_{11} = \mathbf{U}\mathbf{C}\mathbf{U}^{\mathrm{T}} \tag{20}$$

Where $\mathbf{C}$ is $m \times m$ diagonal matrix equal to $\mathbf{I} - \mathbf{S}^2\left(\mathbf{S}^2 + \lambda^2\mathbf{I}\right)^{-1}$, thus its diagonal elements satisfy the condition $0 < c_{ii} \leq 1$. If we denoted A_{22} to the lower right block of matrix $\mathbf{A}$, then $\mathrm{A}_{22} = \mathbf{I} - \kappa\mathbf{N}$, and from equation (10) we can write:

$$\mathrm{A}_{22} = \mathbf{I} - \kappa\left(\mathbf{I} - \mathbf{\Gamma}\mathbf{L}\mathbf{\Gamma}^{\mathrm{T}}\right) \tag{21}$$

Since that $\mathbf{\Gamma}\mathbf{\Gamma}^{\mathrm{T}} = \mathbf{I}$ ($\mathbf{\Gamma}$ is orthonormal) then:

$$\mathrm{A}_{22} = \mathbf{\Gamma}\left(\mathbf{I} - \kappa\mathbf{I} + \kappa\mathbf{L}\right)\mathbf{\Gamma}^{\mathrm{T}} \tag{22}$$

Or:

$$\mathrm{A}_{22} = \mathbf{\Gamma}\mathbf{B}\mathbf{\Gamma}^{\mathrm{T}} \tag{23}$$

Where $\mathbf{B}$ is $n \times n$ diagonal matrix, with the diagonal values:

$$b_{ii} = \begin{cases} 1 & i \leq m \\ 1 - \kappa & i > m \end{cases} \tag{24}$$

Thus we can rewrite matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{U}\mathbf{C}\mathbf{U}^{\mathrm{T}} & 0 \\ -\mathbf{J}^{\bigstar} & \mathbf{\Gamma}\mathbf{B}\mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \tag{25}$$

Or:

$$\mathbf{A} = \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{C} & 0 \\ \mathbf{\Omega} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{U}^{\mathrm{T}} & 0 \\ 0 & \mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \tag{26}$$

From the previous equation we notice that the matrix $\mathbf{A}$ is a similarity transform, so we rewrite it as:

$$\mathbf{A} = \mathbf{O}\mathbf{D}\mathbf{O}^{\mathrm{T}} \tag{27}$$

Where, $\mathbf{O}$ is an $(m+n) \times (m+n)$:

$$\mathbf{O} = \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{\Gamma} \end{bmatrix} \tag{28}$$

Since that both matrices $\mathbf{U}, \mathbf{\Gamma}$ are orthonormal, and they are at opposite sides on the diagonal blocks of $\mathbf{O}$, then matrix $\mathbf{O}$ is orthonormal. Also, $\mathbf{D}$ is an $(m+n) \times (m+n)$:

$$\mathbf{D} = \begin{bmatrix} \mathbf{C} & 0 \\ \mathbf{\Omega} & \mathbf{B} \end{bmatrix} \tag{29}$$

Since that $\mathbf{C}$ and $\mathbf{B}$ are both diagonal matrices, then matrix $\mathbf{D}$ is a lower triangular, where its diagonal elements are equal to their counterparts of matrices $\mathbf{C}$ and $\mathbf{B}$. By substituting (27) in the dynamics equation (18), we find:

$$\boldsymbol{u}_{k+1} = \mathbf{O}\mathbf{D}\mathbf{O}^{\mathrm{T}}\boldsymbol{u}_k \tag{30}$$

Or

$$\mathbf{O}^{\mathrm{T}}\boldsymbol{u}_{k+1} = \mathbf{D}\mathbf{O}^{\mathrm{T}}\boldsymbol{u}_k \tag{31}$$

For the system to be stable, the diagonal elements (eigenvalues) of the lower triangular matrix $\mathbf{D}$ shall be from zero

to one, this is already satisfied for the diagonal elements of the sub-matrix $\mathbf{C}$. Thus, by analysing the diagonal elements of sub-matrix $\mathbf{B}$, equation (24), we conclude that the system is stable when the following condition is satisfied:

$$0 \leq 1 - \kappa \leq 1 \tag{32}$$

Or the system is stable when κ is from zero to one.

IV. Experimental tests and results

In this section we compare our proposed scheme with the legacy Damped Least Squares algorithm 1 in terms of cyclicity performance. The tests were performed in simulations and on a real robot. For the simulations, MATLAB was used for performing the IK calculations, and Vrep was merely used for showing results (3D graphics). The same robot in Figure 1 with 9 DoF is used for the simulations. However, Vrep own IK module was disabled.

Then, after showing the results in simulation we demonstrate the effectiveness of the scheme for generating controlled cyclic motions on real KUKA iiwa (redundant) robot performing a task in Cartesian space. Also, we discuss the advantage of the proposed scheme for solving a problem we encountered due to non-cyclic behaviour on a real KUKA iiwa performing 3D printing.

A. Test 1

In this test, the EEF of the 9 DoF robot in the simulation shown in Figure 1 moves on a circular path with radius of 200 mm, while keeping the EEF orientation fixed during the motion. To calculate the inverse kinematics (joints angles), the proposed scheme and the legacy damped least squares in algorithm 1 are used. For a fair comparison (to achieve an identical update rate for both implementations), we did not use Vrep's internal IK (DLS) calculation module. Rather, we implemented both algorithms in Matlab. For both cases the update rate is around 200Hz, damping factor $\lambda^2 = 0.1$ and $\varepsilon = 1e-15$, while for the proposed scheme the value of κ is 0.5. During the simulation, data is collected and plotted afterwards. Figure 3 shows the results for the joints angles of the legacy DLS method, the non-cyclic behaviour of the robot is evident (it is noticed that both DLS solutions using Vrep solver Figure 2 and our Matlab implementation are quite similar). On the other hand, Figure 4 (a) shows the joint angles plot with time for the proposed scheme, it is noticed that the joint angles are behaving almost cyclically.

The 9 DoF robot used in the simulations is a modified version of the KUKA iiwa with two more extra degrees of freedom. In the Kuk iiwa 7 DoF robot, the third joint is considered as the redundant axis of the robot. Given that our robot is modified from KUKA iiwa by adding two extra joints after the third joint, so we consider the third, and the two newly added joints (fourth and fifth) as the redundant axes. For evaluation of the cyclic behaviour of the proposed scheme, we plot the angle evolution with time for the third, fourth and fifth (redundant) joints in Figures 4 (b), 4 (c) and 4 (d) respectively. Also, the plots contain vertical lines marking the end of each cycle (when the EEF completes a circle). It is noticed that the behaviour of the all joints including the redundant is almost cyclic. Where each cycle of EEF motion (two seconds), it is noticed that joint four of the robot returns back very close to its initial position $-\frac{\pi}{3}$ with a small error of $0.135e-3$ radians, and the third and the fifth joints are also returning back to their initial position at 0 degrees at the end of each cycle of EEF motion with a small error of $3.2e-3$ and $2.2e-3$ radians respectively.

To assess the accuracy of the proposed scheme in tracking the EEF motion, the errors in the position/orientation of the resulting EEF motion from the reference (tracked) path are calculated and plotted in Figure 4 (e) (for positioning errors) and Figure 4 (f) (for orientation error). From the results it is concluded that the proposed method offers an accurate joint angles calculation.

B. Test 2

A real KUKA iiwa 14R820 robot was used to test the proposed algorithm and to compare its performance with legacy DLS method. In the test, the robot was controlled from MATLAB using the KUKA Sunrise Toolbox [30] by streaming the joints angles to the controller, which are calculated using the proposed scheme and the legacy DLS method. The robot starts form an initial configuration listed in Table I. Then, the EEF is moved on a circle in a plane parallel to the YZ plane of the robot base with a radius of 150 mm while keeping a fixed orientation of the EEF during the motion, Figure 5. We note here, that for performing this motion, a redundant robot is required, where a 6 DoF robot suffers singularity for the proposed path (this was confirmed in simulation by fixing the redundant axis - joint three - of the KUKA iiwa and trying to move the robot on the path - using Jacobian inverse, where suddenly the non-redundant robot gets into a singular configuration, causing discontinuity in joints motion). However, using the redundancy (all 7 axes) the legacy DLS method was able to track EEF motion, but at the end of the motion the robot deviates from its initial configuration as shown in the joint angles plot in Figure 6, this is also evident in the video of the experiment which is available in the multimedia material. On the other hand, using our proposed scheme, the robot is successfully tracking the EEF motion, in addition it returns back to its initial configuration at the end of the motion, this is shown in the joint angles plot in Figure 7, also in the video in the multimedia material.

To measure the accuracy of the motion on the real industrial robot, a laser tracker was used to track the EEF motion with a micrometer accuracy. The laser tracker from API is shown in lower left side of Figure 5, it is tracking the motion of a laser reflector mounted at the robot flange at a 100 HZ. From the tracker measurements it is noticed that the EEF motion is almost a circle parallel to the YZ plane of the robot base, the best fit radius for the EEF circular motion is 150.13 mm, which differs slightly from the theoretical radius in our program which is 150mm. Also, the maximum/minimum radial error away from the best fit is ± 0.2 mm , a plot showing the radial error (magnified $\times 80$) is shown in Figure 8. On the other

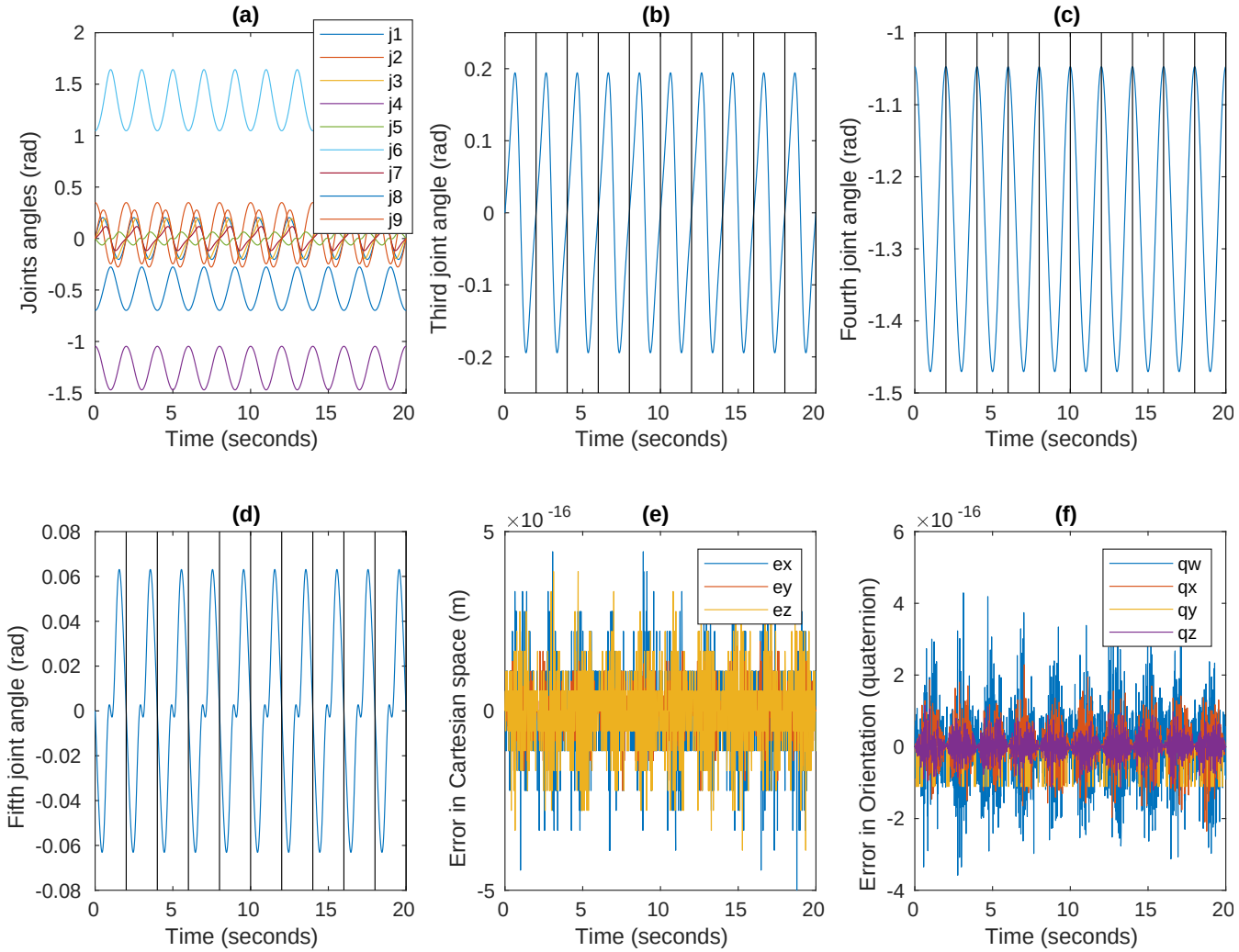


Fig. 4

TEST 1: SIMULATION RESULTS FOR THE PROPOSED SCHEME WHILE TRACKING EEF MOTION WITH 9 DOF ROBOT. THE SUB-FIGURES ARE: (A) JOINT ANGLES PLOT, (B) THIRD JOINTS ANGLE PLOT, (C) FORTH JOINT ANGLE PLOT, (D) FIFTH JOINT ANGLE PLOT, (C) POSITIONING TRACKING ERROR, (D) ORIENTATION TRACKING ERROR.

TABLE I

JOINT ANGLES AT THE INITIAL CONFIGURATION FOR THE KUKA IIWA14R820 ROBOT IN TEST 2.

Joint	J1	J2	J3	J4	J5	J6	J7
Angle	0	$\frac{\pi}{6}$	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{6}$	0

hand, the X coordinate errors during the motion are between ± 0.15mm. It is noticed that the error in the motion are bigger than the theoretical accuracy of the robot given in the data-sheet which is ± 0.1 mm, this is attributed to various factors, including:

1) The difference in the theoretical DH parameters listed in the data sheets from the actual DH parameter of the real robot;
2) Tracking error for the real time servo control loop of the joints;
3) Other factors including, thermal errors, gear inaccuracies and backlash which aggravates by continuous usage (the robot used in the test is not brand new).

However, the results show that the proposed scheme is valid for controlling redundant industrial robots, at the same time, successfully achieving a controlled cyclic solution and preserving immunity to singularities while tracking EEF motion.

C. Use Case

While using KUKA iiwa (7 DoF) robot for 3D printing application Figure 9, the authors first used the legacy DLS method for controlling the robot on the path. The idea was to take advantage of the redundancy to avoid any singular configuration while performing the 3D printing, which is the

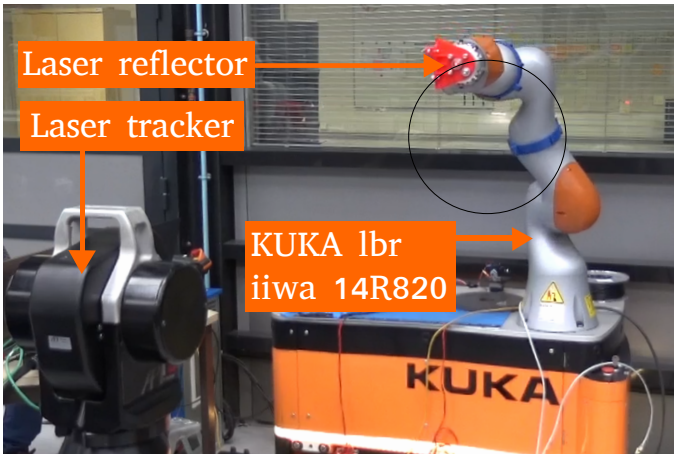


Fig. 5

TEST 2: KUKA IIWA IS MOVING ON A CIRCULAR PATH PARALLEL TO YZ PLANE. A LASER TRACKER IS USED TO MEASURE THE MOTION AT THE EEF OF THE ROBOT WHILE MOVING ON THE CIRCULAR PATH.

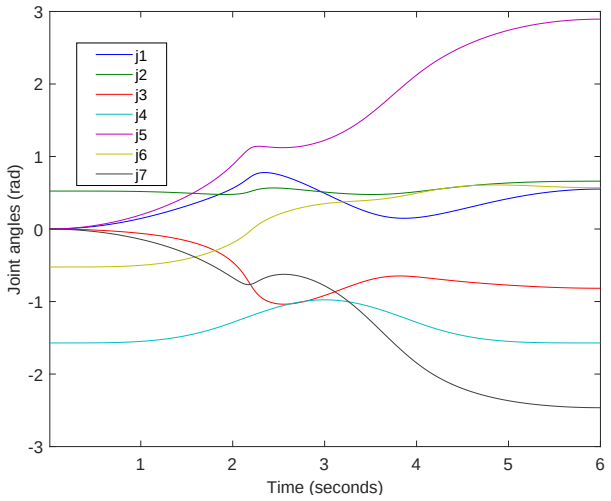


Fig. 6

TEST 2: JOINT ANGLES OF KUKA IIWA14R820 WHILE TRACKING EEF MOTION USING LEGACY DLS METHOD.

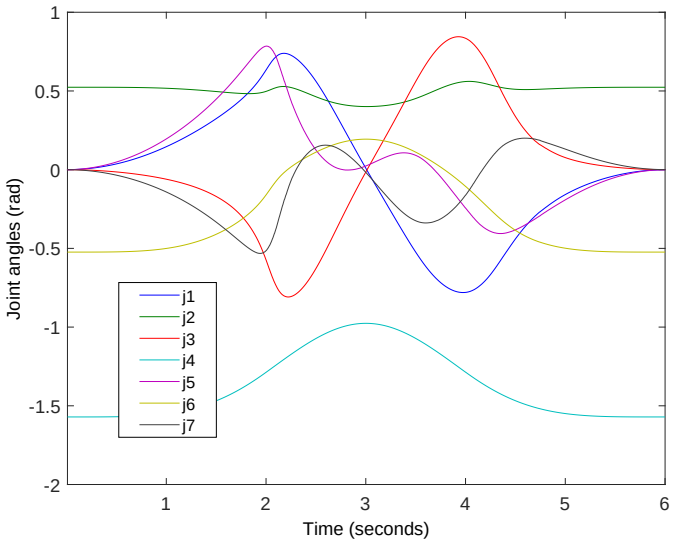


Fig. 7

TEST 2: JOINT ANGLES OF KUKA IIWA14R820 WHILE TRACKING EEF MOTION USING PROPOSED SCHEME.

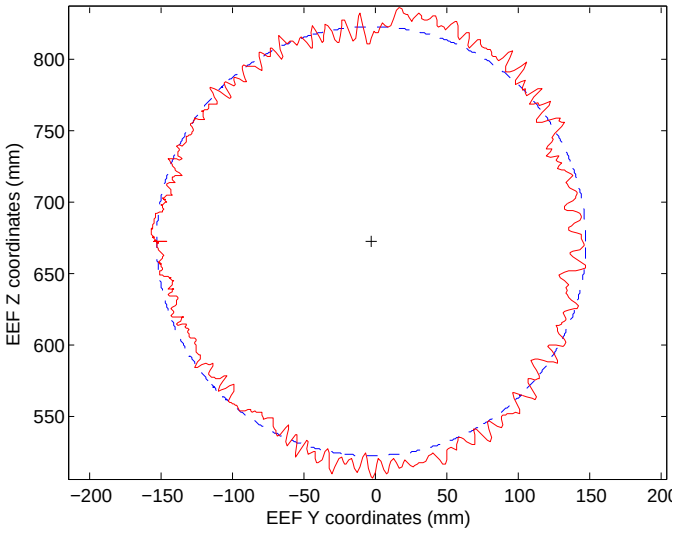


Fig. 8

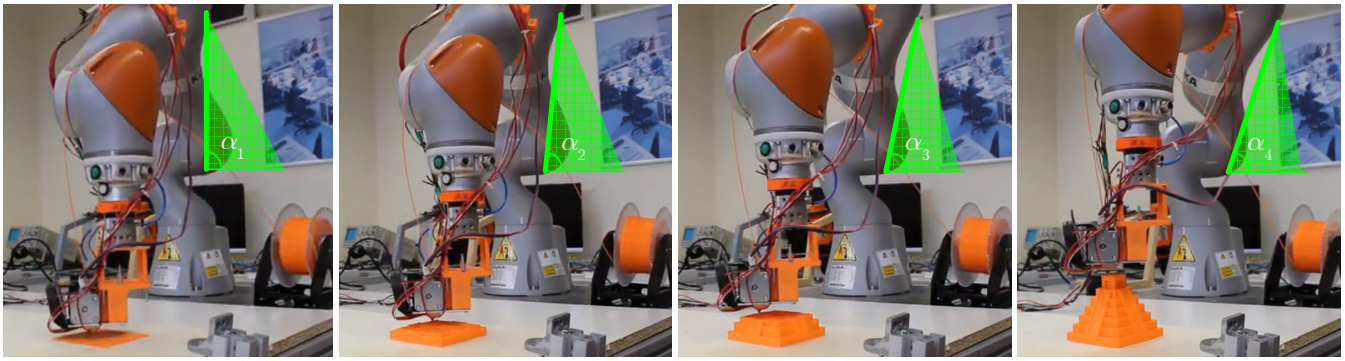
TEST 2: THE RADIAL ERROR MAGNIFIED (X80) IN EEF MOTION AS MEASURED BY THE LASER TRACKER.

main advantage for the DLS method for IK calculation. Unfortunately, the non-cyclicity issue of DLS method was invoking another phenomena, with time the swivel angle [31, 32] of the elbow was drifting, shown in Figure 9, which caused concerns of the possibility that the robot elbow colliding with surrounding structure around the robot, or even hitting the table if a full inversion occurred of the robot elbow. The simple solution of stopping the robot to readjust the swivel angle was not an option since it will affect the printing quality. On the other hand, the proposed method proved efficient for solving this problem (swivel angle drift), while at the same time getting all of the advantages of the DLS method including:

1) The use of redundancy for avoiding singular configurations;
2) Very precise tracking of EEF motion which is important for producing quality 3D prints.

V. CONCLUSION

This study addressed the non-cyclic behaviour for the inverse kinematics solution in redundant robots when using the pseudo inverses, particularly the damped least squares method. Using our proposed scheme it is possible to achieve a controlled cyclic solution, consequently, the behaviour of the redundant robot is predictable for repeated tasks, this is very important in many applications especially for industrial robotic applications. Unlike the extended Jacobian method, the proposed scheme does not suffer from singularity issues. It is



(a) (b) (c) (d)

Fig. 9

3D PRINTING WITH KUKA IIWA, THE SWIVEL ANGLE DRIFTING WITH TIME DUE TO NON-CYCLIC PROBLEM OF DLS METHOD.

efficient computationally and gives accurate solution, where it is evident from simulation tests on 9 DoF robot that the error in the calculations for EEF motion tracking is trivial (1e-15). We also tested the validity of the proposed scheme for controlling real industrial robots, laser tracking measurements, and joints angles behaviour during the real test proved the advantages of the proposed scheme. It is concluded that the proposed scheme has various applications spanning robot simulation and industrial robot control.

VI. Acknowledgment

This research was partially supported by Portugal 2020 project DM4Manufacturing POCI-01-0145-FEDER-016418 by UE/FEDER through the program COMPETE 2020, and the Portuguese Foundation for Science and Technology (FCT) SFRH/BD/131091/2017 and COBOTIS (PTDC/EME-EME/32595/2017).

References

[1] Lorenzo Sciavicco and Bruno Siciliano. *Modeling and control of robot manipulators*. Springer-Verlag, London, 2000.

[2] Tadej Petrič and Leon Žlajpah. Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem. *Robotics and Autonomous Systems*, 61(9):948–959, 2013.

[3] R.V. Patel and F. Shadpey. *3 Collision Avoidance for a 7-DOF Redundant Manipulator*, pages 35–78. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[4] M. Safeea, R. Bearee, and P. Neto. Collision avoidance of redundant robotic manipulators using newton's method. *Journal of Intelligent and Robotic Systems*, 2020.

[5] Abdelrahem Atawnih, Dimitrios Papageorgiou, and Zoe Doulgeri. Kinematic control of redundant robots with guaranteed joint limit avoidance. *Robotics and Autonomous Systems*, 79:122–131, 2016.

[6] A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, Dec 1977.

[7] J. Hollerbach and K. Suh. Redundancy resolution of manipulators through torque optimization. *IEEE Journal on Robotics and Automation*, 3(4):308–316, 1987.

[8] Miomir Vukobratovic and Manja Kirćanski. A dynamic approach to nominal trajectory synthesis for redundant manipulators. *IEEE transactions on systems, man, and cybernetics*, (4):580–586, 1984.

[9] C. A Klein and B. E Blaho. Dexterity measures for the design and control of kinematically redundant manipulators. *The international journal of robotics research*, 6(2):72–83, 1987.

[10] Fabrizio Fiacco and Alessandro De Luca. Fast redundancy resolution for high-dimensional robots executing prioritized tasks under hard bounds in the joint space. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2500–2506. IEEE, 2013.

[11] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of intelligent and robotic systems*, 3(3):201–212, 1990.

[12] Daniel E Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 10(2):47–53, 1969.

[13] John Baillieul. Kinematic programming alternatives for redundant manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 722–728. IEEE, 1985.

[14] Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2):3–9, 1985.

[15] Yoshihiko Nakamura and Hideo Hanafusa. Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3):163–171, 09 1986.

[16] Stefano Chiaverini, Giuseppe Oriolo, and Ian D. Walker. *Kinematically Redundant Manipulators*, pages 245–268. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[17] Charles A Klein and Ching-Hsiang Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):245–250, 1983.

[18] Wisama Khalil and Etienne Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004.

[19] Zhi-Jun ZHANG and Yu-Nong ZHANG. Equivalence of different-level schemes for repetitive motion planning of redundant robots. *Acta Automatica Sinica*, 39(1):88 – 91, 2013.

[20] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu. Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators. *IEEE/ASME Transactions on Mechatronics*, 22(3):1423–1434, 2017.

[21] Yunong Zhang and Zhijun Zhang. *Repetitive motion planning and control of redundant robot manipulators*. Springer Science & Business Media, 2014.

[22] Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):93–101, 1986.

[23] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE, 2013.

[24] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[25] Tsing-Hua Chen, Fan-Tien Cheng, York-Yih Sun, and Min-Hsiung Hung. Torque optimization schemes for kinematically redundant manipulators. *Journal of Robotic Systems*, 11(4):257–269, 1994.

[26] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.

[27] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer Science & Business Media, 2011.

[28] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[29] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[30] M. Safeea and P. Neto. Kuka sunrise toolbox: Interfacing collaborative robots with matlab. *IEEE Robotics Automation Magazine*, 26(1):91–96, March 2019.

[31] H. Su, W. Qi, C. Yang, A. Aliverti, G. Ferrigno, and E. De Momi. Deep neural network approach in human-like redundancy optimization for anthropomorphic manipulators. *IEEE Access*, 7:124207–124216, 2019.

[32] Y Wang and P Artemiadis. Closed-form inverse kinematic solution for anthropomorphic motion in redundant robot arms. *Adv Robot Autom*, 2(110):2, 2013.