NONLINEAR ENGINEERING AND ROBOTICS

MSC 2010: 74S30

Stiffness Model Reduction for Manipulators with Double Encoders: Algebraic Approach

S. K. Mikhel, A. S. Klimchik

The accuracy of the robot positioning during material processing can be improved if the deformation under the load is taken into account. A manipulator stiffness model can be obtained using various approaches which differ in the degree of detail and computational complexity. Regardless of the model, its practical application requires knowledge of the stiffness parameters of the robot components, which implies solving the identification problem.

In this work, we consider a reduced stiffness model, which assumes that the manipulator links are rigid, while the joints are compliant and include both elasticities in the joints themselves and the elastic properties of the links. This simplification reduces the accuracy of the model, but allows us to identify the stiffness parameters, which makes it suitable for practical application. In combination with a double encoders measurement system, this model allows for real-time compensation of compliance errors, that is, the deviation of the real end-effector position from the calculated one due to the deformation of the robot under load.

The paper proposes an algebraic approach to determining the parameters of the reduced model in a general form. It also demonstrates several steps that can be done to simplify computations. First, it introduces the backward semianalytical Jacobian computation technique, which allows reducing the number of operations for the manipulator with virtual joints. Second, it provides an algorithm for the calculation of the required intermediate matrices without explicit Jacobian calculation and using more compact expressions.

To compare the proposed techniques with the experimental approach, the robot deformation under load is simulated and the tool displacement is estimated. It is shown that both approaches are equivalent in terms of accuracy. While the experimental method is easier to implement, the algebraic approach allows analyzing the contribution of each link in a reduced model of elasticity. Besides, the obtained estimate of the parameters does not depend on the accuracy of measurements and configurations used in the identification process.

Keywords: robot stiffness, Jacobian evaluation, identification

Received July 30, 2021
Accepted August 25, 2021

Stanislav K. Mikhel
s.mikhel@innopolis.ru
Alexandr S. Klimchik
a.klimchik@innopolis.ru
Innopolis University,
ul. Universitetskaya 1, Innopolis, 420500 Russia

1. Introduction

The real-world manipulators are not perfectly rigid and can deform under different forces applied by external objects. The amplitude of such deformation depends on the values of the external force and stiffness of the robot elements, which sometimes can be ignored, but in many applications it should be taken into account. In particular, the actual tool trajectory of industrial manipulators that deal with high payloads can be significantly different from the desired path points. In the case of collaborative robots, the stiffness of joints can be reduced to make the robot safer. For such tasks, the end-effector trajectory can be less important, but we usually need information about the force and location of the collision.

Double encoders are a sensor system for joint deflection measurement. Its schematic representation is shown in Fig. 1. The system consists of two encoders, one is on the load side and the other is on the motor side of the shaft. Under the assumption of negligible backlash and constant friction, the difference between encoder angles depends on joint stiffness and applied force. Thus, the information obtained can be useful in both cases, for the estimation of the tool deflection and the force value.

Fig. 1. Double encoders scheme

There are some studies on the use of double encoders for various robotics tasks. The analysis of the relationship between information from primary and secondary encoders and its application to compliance error compensation can be found in [8]. It was shown that the stiffness model, in combination with primary and secondary encoder values, can be used to compensate up to 80 % of the tool deflection. The algorithm of the reduced elastostatic model calibration and compensation strategy is described in [11]. Some robot manufacturers try to integrate double encoders into the joints of a manipulator and provide control algorithms for compensation of the joint compliances [6, 19].

External force detection via double encoders can be found in [4] and [13]. In both cases, it is assumed that Hooke's law holds and deflection value is proportional to an applied external force in nonsingular configurations. However, the former paper offers to evaluate external impact as a simple difference between the measured torque and model prediction, while the latter tends to use some kind of observer, as is done, for example, in [2, 12].

Robot control is another area where double encoders can be useful. In particular, the self-resonance cancellation technique allows overcoming the problem of antiresonance restriction in the two-inertia system with the help of a virtual angle. This angle is calculated from motor and load side positions and provides a lack of resonance [20]. A combination of such an approach with the traditional P-PI control technique allows one to eliminate some other faults, the vibration and disturbance suppression performances are improved and the control bandwidth becomes higher [5].

In this paper, we focus on compliance error compensation for a serial manipulator. To be more specific, we propose an algebraic way of estimating the reduced stiffness model based on the link parameters and kinematic robot structure. Combination of such a model with the double encoder measurement information can be used for real-time compensation of the manipulator

deformations under the external load. The remainder of the paper is organized as follows. Section 2 defines full and reduce stiffness models. Section 3 demonstrates an algebraic approach and shows the ways of its simplification. Section 4 demonstrates stiffness identification for the simple robot model using algebraic and experimental approaches. Section 5 summarizes the main contributions of the paper.

2. Stiffness modeling

2.1. Full model

The full stiffness model allows us to describe the effect of stiffness in joints and links on the end-effector deflection under the applied force in the form of a matrix. Three main approaches can be used for the manipulator stiffness modeling: the finite element analysis (FEA), the matrix structural analysis (MSA), and the virtual joint method (VJM). The FEA technique is the most accurate of them since it allows one to present components with their true shape and dimension [17]. However, it involves high computational expenses. The MSA method incorporates the main ideas of the FEA, but works with 3-dimensional flexible beams [3, 9]. It reduces the computational effort, but the size of matrices is still high, especially for the analytical approach. The VJM technique is based on an extension of the traditional rigid model by adding virtual joints to describe elastic deformations of the real joints and links [1]. It provides a reasonable trade-off between the model accuracy and computational complexity and will be further used in this paper. This method can be used not only for industrial manipulators, but also for other robot types, for example, in anthropomorphic robots [18].

Assume that the robot is serial, each elastic element can be replaced with a combination of a rigid analog and a virtual spring for all possible elastic deformations (Fig. 2). In this case, the transform matrix $\mathbf{H}_{tot}$ of the robot can be found as

$$\mathbf{H}_{tot} = \mathbf{H}_{base} \prod_{i=1}^N [\mathbf{R}_i(q_i + \theta_i^{Ac}) \mathbf{T}_i \mathbf{H}_{3D}(\boldsymbol{\theta}_i^{3D})] \mathbf{H}_{tool}, \quad (2.1)$$

where $\mathbf{T}$ and $\mathbf{R}$ are homogeneous matrices of translation and rotation, $\mathbf{q}$ is a vector of joint angles, $\boldsymbol{\theta}^{Ac}$ and $\boldsymbol{\theta}^{3D}$ are virtual joint coordinates for active joint and link, respectively, $\mathbf{H}_{base}$ and $\mathbf{H}_{tool}$ are homogeneous transformation matrices for base and tool, respectively. $\mathbf{H}_{3D}$ denotes the deflection in the link, caused by translations and rotations of all the virtual joints:

$$\mathbf{H}_{3D}(\boldsymbol{\theta}^{3D}) = \mathbf{T}_x(\theta_x) \mathbf{T}_y(\theta_y) \mathbf{T}_z(\theta_z) \mathbf{R}_x(\theta_\alpha) \mathbf{R}_y(\theta_\beta) \mathbf{R}_z(\theta_\gamma).$$

Here θ_j defines the angle of virtual joint j , subscripts α , β and γ correspond to rotation about axes x , y and z , respectively.

Model (2.1) allows one to obtain a function $\mathbf{t}(\mathbf{q}, \boldsymbol{\theta})$ that returns the vector with 6 elements (3 for position and 3 for orientation of the end-effector) based on the current configuration and the virtual joint state. The observation Jacobian with respect to virtual joint angles can be found as $\mathbf{J} = \partial \mathbf{t}(\mathbf{q}, \boldsymbol{\theta}) / \partial \boldsymbol{\theta}$. Thus, the stiffness matrix in Cartesian space is equal to [7]:

$$\mathbf{K}_C = (\mathbf{J} \mathbf{K}_\theta^{-1} \mathbf{J}^T)^{-1},$$

where $\mathbf{K}_\theta$ is the stiffness matrix in the joint space. The matrix $\mathbf{K}_\theta$ is diagonal with scalar values in the case of joints and 6×6 matrices for links. If any external force $\mathbf{w}$ is applied to the end-effector, the amount of deflection $\Delta \mathbf{t}$ is equal to

$$\Delta \mathbf{t} = \mathbf{K}_C^{-1} \mathbf{w}. \quad (2.2)$$

Fig. 2. VJM technique for manipulator (white elements are rigid, colored are elastic)

The expression (2.2) provides the most complete description of the manipulator deformation within the VJM model. Further, it will be used as ground truth in comparison to the experimental and algebraic approaches.

2.2. Reduced model

The full stiffness model (2.2) requires knowledge of all the virtual joint stiffness parameters. For the N link manipulator its number is at most $N \cdot (6 \times 6 + 1)$. Besides, such a model depends on joints and links and cannot be predicted based on the joint displacements only. Thus, we need a reduced model that would be easy to identify and use in combination with double encoders.

The reduced stiffness model assumes that the robot links are rigid, while the joints are compliant. In this case, the effect of link elasticity is expressed by changing the initial joint stiffness coefficients. Equation (2.2) then becomes

$$\Delta \mathbf{t} = (\mathbf{J} \mathbf{k}_{red} \mathbf{J}^T) \mathbf{w}, \quad (2.3)$$

where $\mathbf{k}_{red}$ is the diagonal compliance matrix for the robot joints.

In practice, the reduced stiffness model parameters can be found using the elastostatic calibration. To identify them we should represent (2.3) in the form

$$\Delta \mathbf{t} = \mathbf{A} \mathbf{k},$$

where $\mathbf{A}$ is an observation matrix, $\mathbf{k} = \text{vec}(\mathbf{k}_{red})$ is a vector representation of the compliance matrix. The observation matrix can be found as

$$\mathbf{A} = [\mathbf{J}_1 \mathbf{J}_1^T \mathbf{w}, \mathbf{J}_2 \mathbf{J}_2^T \mathbf{w}, \dots, \mathbf{J}_N \mathbf{J}_N^T \mathbf{w}]$$

using the columns of Jacobian $\mathbf{J} = [\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_N]$.

The process of identification consists of a series of measurements of the end-effector deflections under the influence of the applied force in different joint configurations. The estimated value of joint compliance could be found, for example, with the help of the least-squares technique:

$$\hat{\mathbf{k}} = \left(\sum_{j=1}^m \mathbf{A}_j^T \mathbf{A}_j \right)^{-1} \left(\sum_{j=1}^m \mathbf{A}_j^T \Delta \mathbf{t}_j \right), \quad (2.4)$$

where m is the number of experiments.

The compliance of the reduced model (2.4) can be used as follows. Assume that $\mathbf{q}_{goal}$ is a trajectory point in joint space and $\Delta\mathbf{q}$ is the measured or estimated vector of differences in double encoders. Then the corrected joint state $\mathbf{q}_{corr}$ is equal to

$$\mathbf{q}_{corr} = \mathbf{q}_{goal} - \mathbf{k}_{corr}\Delta\mathbf{q},$$

where $\mathbf{k}_{corr} = \text{diag}(\hat{\mathbf{k}})\mathbf{K}_q$ and $\mathbf{K}_q$ is the stiffness matrix of the nonvirtual joints.

3. Algebraic approach

3.1. Parameter estimation

Let us consider another technique for estimating the model parameters [14]. The reduced model should provide a Cartesian stiffness matrix equal to the full model. In terms of compliance, it can be written as

$$\mathbf{J}\mathbf{k}_{red}\mathbf{J}^T \approx \mathbf{J}\mathbf{k}_\theta\mathbf{J}^T, \quad (3.1)$$

where $\mathbf{k}_\theta = \mathbf{K}_\theta^{-1}$ is the compliance matrix of the full model. For further analysis it is convenient to transform both sides of (3.1) into the form $\text{vec}(\mathbf{J}\mathbf{k}\mathbf{J}^T) = \mathbf{B}\text{vec}(\mathbf{k})$, where $\text{vec}(\cdot)$ is the operator of vectorization, $\mathbf{B} = \mathbf{J} \otimes \mathbf{J}$ can be obtained with the help of Kronecker product. The vector $\text{vec}(\mathbf{k}_{red} - \mathbf{k}_\theta)$ can be divided into two subvectors: $\boldsymbol{\delta}$ corresponds to joint elements and is equal to the difference between the reduced and full models, $\mathbf{k}_L$ corresponds to link compliance elements. Similarly, the matrix columns can be grouped into the joint $\mathbf{B}_J$ and link $\mathbf{B}_L$ submatrices. Then the problem (3.1) is formulated as follows: find such a joint compliance deflection $\boldsymbol{\delta}$ that compensates link compliance in the best way, i. e.,

$$\boldsymbol{\delta} = \underset{\mathbf{q} \in \mathbf{Q}}{\text{argmin}} \|\mathbf{B}_J(\mathbf{q})\boldsymbol{\delta} - \mathbf{B}_L(\mathbf{q})\mathbf{k}_L\|,$$

where $\mathbf{Q}$ is the space of all available robot configurations $\mathbf{q}$.

The vector $\boldsymbol{\delta}$ can be found from the equation

$$\frac{d}{d\boldsymbol{\delta}} \int_{\mathbf{Q}} ((\mathbf{B}_J(\mathbf{q})\boldsymbol{\delta} - \mathbf{B}_L(\mathbf{q})\mathbf{k}_L)^T (\mathbf{B}_J(\mathbf{q})\boldsymbol{\delta} - \mathbf{B}_L(\mathbf{q})\mathbf{k}_L)) d\mathbf{q} = \mathbf{0}.$$

The solution can be expressed in the form

$$\mathbf{M}_{JJ}^{int}\boldsymbol{\delta} = \mathbf{M}_{JL}^{int}\mathbf{k}_L, \quad (3.2)$$

where

$$\mathbf{M}_{JJ}^{int} = \int_{\mathbf{Q}} \mathbf{M}_{JJ}(\mathbf{q}) d\mathbf{q}, \quad \mathbf{M}_{JL}^{int} = \int_{\mathbf{Q}} \mathbf{M}_{JL}(\mathbf{q}) d\mathbf{q}$$

and

$$\begin{aligned} \mathbf{M}_{JJ}(\mathbf{q}) &= \mathbf{B}_J^T(\mathbf{q})\mathbf{B}_J(\mathbf{q}), \\ \mathbf{M}_{JL}(\mathbf{q}) &= \mathbf{B}_J^T(\mathbf{q})\mathbf{B}_L(\mathbf{q}). \end{aligned}$$

Thus, the joint stiffness estimation for the reduced model is equal to

$$\hat{\mathbf{k}} = \text{vec}(\mathbf{k}_\theta) + \boldsymbol{\delta},$$

which is the algebraic alternative to the experimental approach (2.4).

The matrices $\mathbf{M}_{JJ}$ and $\mathbf{M}_{JL}$ can be found without computation of the $\mathbf{B}$ matrix. To show it, let us regroup the Jacobian columns into blocks:

$$\mathbf{J} = [\mathbf{J}^{jnt} \mid \mathbf{J}^{lnk,1} \mid \mathbf{J}^{lnk,2} \mid \dots],$$

where the submatrix $\mathbf{J}^{jnt}$ contains all the columns that correspond to the robot joints and the submatrix $\mathbf{J}^{lnk,i}$ contains elements for the virtual springs in link i . The same can be done with the matrix $\mathbf{B} = [\mathbf{B}_J \mid \mathbf{B}_L^{lnk,1} \mid \mathbf{B}_L^{lnk,2} \mid \dots]$, therefore $\mathbf{M}_{JJ} = \mathbf{B}_J^T \mathbf{B}_J$ and $\mathbf{M}_{JL} = [\mathbf{B}_J^T \mathbf{B}_L^{lnk,1} \mid \mathbf{B}_J^T \mathbf{B}_L^{lnk,2} \mid \dots] = [\mathbf{M}_{JJ}^{lnk,1} \mid \mathbf{M}_{JJ}^{lnk,2} \mid \dots]$. Since the n th column of $\mathbf{B}_J$ consists only of the elements from the n th column in $\mathbf{J}^{jnt}$, the $\mathbf{M}_{JJ(i,j)}$ matrix element can be found as

$$\mathbf{M}_{JJ(i,j)} = (\mathbf{J}_i^{jnt} \cdot \mathbf{J}_j^{jnt})^2, \quad (3.3)$$

where $\mathbf{J}_n^{jnt}$ is the n th column and “ $\cdot$ ” is a scalar product operation.

The same approach applies to the matrix $\mathbf{M}_{JL}$. Let us consider the vector $\mathbf{k}_L^{lnk,n}$ of the n th link and assume that the j th element of this vector corresponds to $\mathbf{k}_{\theta(p,q)}^{lnk,n}$ in the appropriate joint compliance matrix. The effect of this parameter on the compliance of the i th joint in the reduced model is equal to

$$\mathbf{M}_{JL(i,j)}^{lnk,n} = (\mathbf{J}_i^{jnt} \cdot \mathbf{J}_p^{lnk,n})(\mathbf{J}_i^{jnt} \cdot \mathbf{J}_q^{lnk,n}). \quad (3.4)$$

Equations (3.3) and (3.4) allow finding matrices directly from the Jacobian, which significantly simplifies the required computations.

3.2. Backward semianalytical Jacobian calculation

The algebraic parameter estimation approach requires that the product of the Jacobian columns be integrated. This task is time-consuming and depends on the efficiency of the Jacobian calculation. It can be found, for example, using the geometrical approach [21], the semianalytical method [15] or the product of exponents [10]. In this section we would like to demonstrate a technique that allows one to reduce the computational cost of the Jacobian computation in the case of VJM analysis of a serial manipulator.

The proposed method is based on the semianalytical approach and uses the fact that the derivative of the manipulator homogeneous matrix for the i th joint angle is equal to

$$\frac{\partial \mathbf{H}}{\partial q_i} = \begin{pmatrix} \mathbf{S}_i \mathbf{R}_{ee} & \partial \mathbf{p}_{ee} / \partial q_i \\ 0 & 0 \end{pmatrix},$$

where $\mathbf{S}_i$ is the skew-symmetric matrix for angular velocity terms, and $\mathbf{R}_{ee}$ and $\mathbf{p}_{ee}$ are the end-effector rotation matrix and the position vector, respectively. Let us define the product of the matrix derivative for change in the n th joint angle and inverted end-effector rotation as

$$d\mathbf{H}_n = \frac{\partial \mathbf{H}}{\partial q_n} \mathbf{H}_{rot}^{-1} = \mathbf{a}_n \mathbf{H}'_n \mathbf{b}_n, \quad (3.5)$$

where

$$\mathbf{a}_n = \prod_{i=1}^{n-1} \mathbf{H}_i, \quad \mathbf{H}' = \frac{\partial \mathbf{H}_n}{\partial q_n}$$

$$\mathbf{b}_n = \left(\prod_{j=n+1}^N \mathbf{H}_j \right) \mathbf{H}_{rot}^T, \quad \mathbf{H}_{rot} = \begin{pmatrix} \mathbf{R}_{ee} & 0 \\ 0 & 1 \end{pmatrix}.$$

The n th Jacobian column can be found as

$$[d\mathbf{H}_n^{(1,4)}, d\mathbf{H}_n^{(2,4)}, d\mathbf{H}_n^{(3,4)}, d\mathbf{H}_n^{(3,2)}, d\mathbf{H}_n^{(1,3)}, d\mathbf{H}_n^{(2,1)}]^T, \quad (3.6)$$

where the first half of elements corresponds to translation and the last half to rotation.

First, we can regroup multipliers in (3.5) to get the derivative-dependent matrix at the end of the equation:

$$\mathbf{a}_n \mathbf{H}'_n \mathbf{b}_n = (\mathbf{a}_n \mathbf{H}_n \mathbf{b}_n) \mathbf{D}_n. \quad (3.7)$$

The bracketed expression is equal to the product of total manipulator transformation matrices with the inverted end-effector orientation $\mathbf{a}_n \mathbf{H}_n \mathbf{b}_n = \mathbf{H} \mathbf{H}_{rot}^{-1}$, it is constant for the current configuration and can be found only once. Thus, the Jacobian elements for the n th column depend on the matrix $\mathbf{D}_n$, which is equal to

$$\mathbf{D}_n = \mathbf{b}_n^{-1} \mathbf{H}_n^{-1} \mathbf{H}'_n \mathbf{b}_n. \quad (3.8)$$

It can be found that for all elementary transformations the product $\mathbf{H}^{-1} \mathbf{H}'$ is equal to zero matrices with one (for translation) or two (for rotation) nonzero elements and empty last row. Hence, the last column of $\mathbf{D}_n$ is zero. On the other hand, the product $\mathbf{H} \mathbf{H}_{rot}^{-1}$ has a rotational part equal to the identity matrix. As a result, equation (3.7) becomes $\mathbf{a}_n \mathbf{H}'_n \mathbf{b}_n = \mathbf{D}_n$ and the Jacobian depends on $\mathbf{D}_n$ only.

The form of the matrix $\mathbf{H}^{-1} \mathbf{H}'$ makes it possible to define the Jacobian as a function of $\mathbf{b}_n$. Suppose that the matrix can be written as

$$\mathbf{b}_n = \begin{pmatrix} r_{x1} & r_{x2} & r_{x3} & p_x \\ r_{y1} & r_{y2} & r_{y3} & p_y \\ r_{z1} & r_{z2} & r_{z3} & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Consider the rotation for the X -axis as an example. After calculation of (3.8) and application of the rule (3.6) the Jacobian column consists of the elements

$$\begin{pmatrix} p_y r_{z1} - p_z r_{y1} \\ p_y r_{z2} - p_z r_{y2} \\ p_y r_{z3} - p_z r_{y3} \\ r_{y2} r_{z3} - r_{y3} r_{z2} \\ r_{y3} r_{z1} - r_{y1} r_{z3} \\ r_{y1} r_{z2} - r_{y2} r_{z1} \end{pmatrix} = \begin{pmatrix} p_y \mathbf{r}_z - p_z \mathbf{r}_y \\ \mathbf{r}_y \times \mathbf{r}_z \end{pmatrix} = \begin{pmatrix} p_y \mathbf{r}_z - p_z \mathbf{r}_y \\ \mathbf{r}_x \end{pmatrix},$$

where $\mathbf{r}_x = (r_{x1} \ r_{x2} \ r_{x3})^T$, $\mathbf{r}_y = (r_{y1} \ r_{y2} \ r_{y3})^T$, $\mathbf{r}_z = (r_{z1} \ r_{z2} \ r_{z3})^T$, “ $\times$ ” is the cross-product operation. Here we have used the orthonormality of the rotation matrix rows: the cross-product of two rows is equal to the third row. The same steps can be done with other transformations, the results are the following:

$$\begin{aligned} \mathbf{J}_{Tx} &= \begin{pmatrix} \mathbf{r}_x \\ 0 \end{pmatrix}, \quad \mathbf{J}_{Ty} = \begin{pmatrix} \mathbf{r}_y \\ 0 \end{pmatrix}, \quad \mathbf{J}_{Tz} = \begin{pmatrix} \mathbf{r}_z \\ 0 \end{pmatrix}, \\ \mathbf{J}_{Rx} &= \begin{pmatrix} p_y \mathbf{r}_z - p_z \mathbf{r}_y \\ \mathbf{r}_x \end{pmatrix}, \quad \mathbf{J}_{Ry} = \begin{pmatrix} p_z \mathbf{r}_x - p_x \mathbf{r}_z \\ \mathbf{r}_y \end{pmatrix}, \quad \mathbf{J}_{Rz} = \begin{pmatrix} p_x \mathbf{r}_y - p_y \mathbf{r}_x \\ \mathbf{r}_z \end{pmatrix}. \end{aligned} \quad (3.9)$$

The proposed approach can be generalized in Algorithm 1.

Algorithm 1 Backward Semianalytical Jacobian calculation

```

procedure BSA
   $\mathbf{H}_{rot} \leftarrow \prod \mathbf{H}_n$  ▷ multiply rotational matrices
   $\mathbf{b} \leftarrow \mathbf{H}_{rot}^T$  ▷ initialize matrix
  for  $\mathbf{H}_n$  from tool to base do
    if  $\mathbf{H}_n$  corresponds to joint then
      Find  $\mathbf{J}_{col}(\mathbf{b})$  from (3.9)
       $\mathbf{b} \leftarrow \mathbf{H}_n \mathbf{T}$  ▷ update matrix
  Concatenate  $\mathbf{J}_{col}$  ▷ reverse order

```

3.3. Application to the algebraic approach

The proposed Jacobian calculation technique allows simplifying the algebraic parameter estimation method. First of all, the Jacobian is typically calculated for the zero virtual joint angles. In this case, all the Jacobian columns for virtual joints of one link can be obtained from the same homogeneous matrix using recombination of elements (3.9).

Equations (3.9) can be used for further simplification of (3.2). When two rotational matrices $\mathbf{R}^a$ and $\mathbf{R}^b$ are connected by the third matrix $\mathbf{R}^{ab}$ in the form $\mathbf{R}^a = \mathbf{R}^{ab} \mathbf{R}^b$, the scalar product of any two rows $\mathbf{r}_i^a$ and $\mathbf{r}_j^b$ is equal to the corresponding matrix element: $\mathbf{r}_i^a \cdot \mathbf{r}_j^b = \mathbf{R}_{(i,j)}^{ab}$. It can be applied to the scalar product of Jacobian columns. Assume that we have two joints with transformations in v and w directions, the corresponding matrices $\mathbf{b}_v$ and $\mathbf{b}_w$, Jacobian columns $\mathbf{J}_v$ and $\mathbf{J}_w$. Then

$$\mathbf{J}_v \cdot \mathbf{J}_w = -\det(\mathbf{K}), \quad (3.10)$$

where $\det(\cdot)$ is the matrix determinant, and $\mathbf{K}$ can be obtained from the following steps:

1. make the matrix $\mathbf{K}$ equal to rotation from $\mathbf{b}_w$ to $\mathbf{b}_v$;
2. replace the w th column with the vector $\mathbf{p}$ of the matrix $\mathbf{b}_v$ if the joint is rotational or else fill it with zeros;
3. replace the v th row with the vector $\mathbf{p}$ of the matrix $\mathbf{b}_w$ if the joint is rotational or else fill it with zeros;
4. set $\mathbf{K}_{v,w} = -1$.

To demonstrate (3.10), let us consider an example for two joints with rotation about the X and Y axes with the intermediate rotational matrix $\mathbf{R}$:

$$\begin{aligned}
 \mathbf{J}_{Rx}^1 \cdot \mathbf{J}_{Ry}^2 &= (p_y^1 r_z^1 - p_z^1 r_y^1)(p_z^2 r_x^2 - p_x^2 r_z^2) + r_x^1 r_y^2 = \\
 &= (p_y^1 R_{zx} - p_z^1 R_{yx})p_z^2 - (p_y^1 R_{zz} - p_z^1 R_{yz})p_x^2 + R_{xy} = \\
 &= \det \begin{pmatrix} 0 & p_x^2 & p_z^2 \\ p_y^1 & R_{yx} & R_{yz} \\ p_z^1 & R_{zx} & R_{zz} \end{pmatrix} + (R_{yz} R_{zx} - R_{zz} R_{yx}) = -\det \begin{pmatrix} p_x^2 & -1 & p_z^2 \\ R_{yx} & p_y^1 & R_{yz} \\ R_{zx} & p_z^1 & R_{zz} \end{pmatrix}.
 \end{aligned}$$

Algorithm 2 shows how the matrices $\mathbf{M}_{JL}$ and $\mathbf{M}_{JJ}$ can be found without explicit Jacobian computation.

Algorithm 2 Calculation of M_{JL} and M_{JJ}

```

procedure FINDM
  find  $\mathbf{b}$  matrices for all joints
   $\mathbf{M}, \mathbf{M}_{JJ}, \mathbf{M}_{JL} \leftarrow$  empty matrices
  for each real joint  $i$  do                                     ▷ Get intermediate products
    for each joint  $j$  do
       $\mathbf{r}_j \leftarrow -\det(\mathbf{K}_{i,j})$                                      ▷ Make row vector
       $\mathbf{M} \leftarrow \text{addRow}(\mathbf{r}, \mathbf{M})$ 
    for row  $\mathbf{r}$  in  $\mathbf{M}$  do                                           ▷ Find matrix elements
       $i \leftarrow 1$ 
      while  $i \leq \text{length}(\mathbf{r})$  do
        if joint  $i$  is real then
           $\mathbf{r}_{JJ} \leftarrow \text{append}(\mathbf{r}_i^2, \mathbf{r}_{JJ})$                  ▷ Make row for  $\mathbf{M}_{JJ}$ 
           $i \leftarrow i + 1$ 
        else
           $\mathbf{x} \leftarrow \text{outerProduct}(\mathbf{r}_{i:(i+5)}, \mathbf{r}_{i:(i+5)})$ 
           $\mathbf{r}_{JL} \leftarrow \text{append}(\text{vec}(\mathbf{x}), \mathbf{r}_{JL})$              ▷ Make row for  $\mathbf{M}_{JL}$ 
           $i \leftarrow i + 6$ 
       $\mathbf{M}_{JJ} \leftarrow \text{addRow}(\mathbf{r}_{JJ}, \mathbf{M}_{JJ})$ 
       $\mathbf{M}_{JL} \leftarrow \text{addRow}(\mathbf{r}_{JL}, \mathbf{M}_{JL})$ 

```

4. Modeling

The practical application of the proposed techniques can be demonstrated with the help of the model of a compliant manipulator. First, we will find parameters of the reduced stiffness model, then we will estimate their efficiency using the model error. This error can be found as an average difference between the deflections obtained from the full and reduced stiffness models. The modeling was performed in the Maxima computer algebra system, sources can be found in [22].

4.1. Manipulator description

As an example, let us consider the robot shown in Fig. 3. Despite its simplicity, the model corresponds to the most loaded part of a typical industrial manipulator. The model includes compliant links and joints, the parameters are listed in Table 1, they are close to parameters of the industrial manipulator FANUC R-2000iC/165F. The links are assumed to be hollow cylinders, the material is aluminum with Young's modulus $7 \cdot 10^{10}$ N/m², Poisson's ratio 0.349 and density 2699 kg/m³.

Table 1. Manipulator parameters

Index	Length, m	External diameter, m	Internal diameter, m	Joint stiffness, N/rad
1	0.324	0.25	0.16	2×10^5
2	1.075	0.21	0.16	3×10^5
3	1.5	0.14	0.10	10^5

In order to simplify evaluation, we use some observations and assumptions.

1. The robot workspace is symmetric for the Z -axis, thus only one position $q_1 = 0$ could be considered.

Fig. 3. Manipulator model

2. The Cartesian stiffness matrix is symmetric, which reduces the number of rows in $\mathbf{M}_{JL}$.
3. The joint stiffness matrix of a link is symmetric, which reduces the number of columns in $\mathbf{M}_{JL}$.
4. We can detect only the position deflection and for this reason use only the positional part of the Jacobian.
5. Tool influence is defined with deflections t_y and t_z in the Y and Z directions, respectively. It is required for the detection of rotational parameters of the virtual joints in the last link. In the case of one deflection, we lose a component, while with all three deflections the complexity of equations increases due to an increase in the number of terms. Numerical values of these parameters are set to $t_y = t_z = 0.01$.

4.2. Analytical estimation

The Jacobian can be divided into the following parts:

$$\begin{aligned}
 J^{jnt} &= \begin{pmatrix} -t_y & S_1 - d_2 s_2 & S_1 \\ S_2 + d_2 c_2 & 0 & 0 \\ 0 & -S_2 - d_2 c_2 & -S_2 \end{pmatrix}, \\
 J^{lnk,1} &= \begin{pmatrix} 1 & 0 & 0 & 0 & S_2 - d_2 s_2 & -t_y \\ 0 & 1 & 0 & -S_1 + d_2 s_2 & 0 & S_2 + d_2 c_2 \\ 0 & 0 & 1 & t_y & -S_2 - d_2 c_2 & 0 \end{pmatrix}, \\
 J^{lnk,2} &= \begin{pmatrix} c_2 & 0 & s_2 & t_y s_2 & S_1 & -t_y c_2 \\ 0 & 1 & 0 & d_3 s_3 - t_z c_3 & 0 & t_z s_3 + d_3 c_3 \\ -s_2 & 0 & c_2 & t_y c_2 & -S_2 & t_y s_2 \end{pmatrix}, \\
 J^{lnk,3} &= \begin{pmatrix} c_{23} & 0 & s_{23} & t_y s_{23} & t_z c_{23} & -t_y c_{23} \\ 0 & 1 & 0 & -t_z & 0 & 0 \\ -s_{23} & 0 & c_{23} & t_y c_{23} & -t_z s_{23} & t_y s_{23} \end{pmatrix},
 \end{aligned}$$

where $s_i = \sin(q_i)$, $c_i = \cos(q_i)$, $s_{ij} = \sin(q_i + q_j)$, $c_{ij} = \cos(q_i + q_j)$, $S_1 = t_z c_{23} - d_3 s_{23}$, $S_2 = t_z s_{23} + d_3 c_{23}$.

Table 2 contains coefficients of impact for each link compliance parameter to the joint compliance in the reduced model. The values are normalized to the maximum equal to 1.04. The parameters are found from equation (3.2) with the help of rules (3.3) and (3.4). One can see that the link stiffness parameters $k_{xy}^{lnk,1}$, $k_{xz}^{lnk,1}$, $k_{x\alpha}^{lnk,1}$, $k_{x\beta}^{lnk,1}$, $k_{yz}^{lnk,1}$, $k_{y\alpha}^{lnk,1}$, $k_{y\beta}^{lnk,1}$, $k_{y\gamma}^{lnk,1}$, $k_{z\beta}^{lnk,1}$, $k_{z\gamma}^{lnk,1}$, $k_{\alpha\beta}^{lnk,1}$, $k_{\alpha\gamma}^{lnk,1}$, $k_{\beta\gamma}^{lnk,1}$, $k_{xz}^{lnk,2}$, $k_{x\alpha}^{lnk,2}$, $k_{x\beta}^{lnk,2}$, $k_{yz}^{lnk,2}$, $k_{y\alpha}^{lnk,2}$, $k_{y\beta}^{lnk,2}$, $k_{z\gamma}^{lnk,2}$, $k_{\alpha\gamma}^{lnk,2}$, $k_{\beta\gamma}^{lnk,2}$ have no influence on the reduced model compliance, on the other hand, the impact of $k_{xx}^{lnk,1}$, $k_{yy}^{lnk,1}$, $k_{zz}^{lnk,1}$, $k_{\alpha\alpha}^{lnk,1}$, $k_{\beta\beta}^{lnk,1}$, $k_{\gamma\gamma}^{lnk,1}$, $k_{xx}^{lnk,2}$, $k_{yy}^{lnk,2}$, $k_{y\gamma}^{lnk,2}$, $k_{zz}^{lnk,2}$, $k_{z\beta}^{lnk,2}$, $k_{\alpha\alpha}^{lnk,2}$, $k_{\beta\beta}^{lnk,2}$, $k_{\gamma\gamma}^{lnk,2}$, $k_{yy}^{lnk,3}$, $k_{zz}^{lnk,3}$ is significant. Although all parameters of the last link affect the stiffness of the manipulator, for most of them the impact is small. The found stiffness is shown in Table 4.

Table 2. Normalized elements of the product $(M_{JJ}^{int})^{-1}M_{JL}^{int}$.

	Link 1			Link 2			Link 3		
Parameter	δ_1	δ_2	δ_3	δ_1	δ_2	δ_3	δ_1	δ_2	δ_3
k_{xx}	10^{-5}	0.03	0.17	10^{-6}	-0.03	0.25	10^{-5}	0.07	-0.08
k_{xy}	0	0	0	-10^{-3}	10^{-8}	10^{-8}	-10^{-3}	10^{-8}	10^{-8}
k_{xz}	0	0	0	0	0	0	10^{-7}	10^{-3}	-0.01
$k_{x\alpha}$	0	0	0	0	0	0	10^{-5}	10^{-5}	-10^{-4}
$k_{x\beta}$	0	0	0	0	0	0	10^{-7}	10^{-3}	-0.01
$k_{x\gamma}$	-0.01	-10^{-3}	-10^{-3}	-10^{-3}	10^{-3}	-10^{-3}	-10^{-7}	-10^{-3}	10^{-3}
k_{yy}	0.26	-10^{-6}	-10^{-6}	0.26	-10^{-6}	-10^{-6}	0.26	-10^{-6}	10^{-6}
k_{yz}	0	0	0	0	0	0	-10^{-5}	0	0
$k_{y\alpha}$	0	0	0	0	0	0	-0.01	10^{-8}	10^{-7}
$k_{y\beta}$	0	0	0	0	0	0	-10^{-5}	0	0
$k_{y\gamma}$	0	0	0	0.37	-10^{-6}	-10^{-5}	-10^{-5}	0	0
k_{zz}	-10^{-6}	0.03	0.17	10^{-6}	0.1	0.1	0	10^{-6}	0.42
$k_{z\alpha}$	-10^{-7}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-7}	10^{-7}	10^{-7}	0.01
$k_{z\beta}$	0	0	0	10^{-6}	-0.79	1	10^{-9}	10^{-5}	-10^{-4}
$k_{z\gamma}$	0	0	0	0	0	0	-10^{-9}	-10^{-5}	10^{-4}
$k_{\alpha\alpha}$	0.32	10^{-6}	10^{-5}	0.29	10^{-5}	10^{-6}	10^{-5}	0	10^{-5}
$k_{\alpha\beta}$	0	0	0	10^{-3}	-10^{-3}	0.01	10^{-7}	10^{-7}	-10^{-6}
$k_{\alpha\gamma}$	0	0	0	0	0	0	-10^{-7}	-10^{-7}	10^{-6}
$k_{\beta\beta}$	0	0.96	0	0	0	0.96	10^{-9}	10^{-5}	-10^{-5}
$k_{\beta\gamma}$	0	0	0	0	0	0	-10^{-9}	-10^{-5}	10^{-5}
$k_{\gamma\gamma}$	0.96	0	0	0.29	-10^{-6}	10^{-5}	10^{-9}	10^{-5}	-10^{-5}

4.3. Experimental estimation

The efficiency of the experimental approach depends on such factors as the set of robot configurations and the accuracy of measurements. In the case of a real robot, any optimization technique can be used to reduce the number of experiments and increase the efficiency of the process. Since we deal with a model, it is enough to use random configurations and force directions.

During the modeling, we generated random configurations inside the robot workspace, for each configuration we applied force with random direction and magnitude $100N$, estimated the end-effector deflections according to the full stiffness model and calculated the observation matrix $\mathbf{A}$. Table 3 shows that it is enough to have about 50 experiments to minimize the estimation error for the proposed manipulator model. The robot stiffness parameters are found from equation (2.4) and shown in Table 4.

Table 3. Dependence of the error on the number of experiments.

Experiments	25	50	75	100
Error, m	6.70×10^{-5}	5.82×10^{-5}	5.86×10^{-5}	5.83×10^{-5}

Table 4. Reduced model stiffness.

Parameter	Algebraic approach	Experimental approach
Stiffness, N/rad	$[1.78 \ 2.87 \ 0.94] \times 10^5$	$[1.78 \ 2.87 \ 0.94] \times 10^5$
Error, m	5.83×10^{-5}	5.82×10^{-5}
Computing time, s	4.01	0.05

4.4. Discussion

Two methods of parameter estimation were considered: algebraic and experimental. Table 4 shows that, in general, the results obtained are equal. Both techniques have characteristics that define the scope of application.

The algebraic approach allows us to find and analyze the influence of link stiffness parameters on the total model of the manipulator. It can be useful at the stage of robot design. Since the matrices $\mathbf{M}_{JJ}^{int}$ and $\mathbf{M}_{JL}^{int}$ depend on manipulator kinematics only, knowledge about the most significant parameters allows variation of the geometry of links to obtain the desired stiffness. Another benefit is the integral character of the result. While the experimental method covers only part of all possible states and forces, the algebraic approach takes into account the total workspace of the robot, which provides the best approximation of the stiffness parameters. Nevertheless, this technique has some disadvantages. First, it requires knowledge of the link stiffness matrices, which is not always possible in practice. Second, $\mathbf{M}_{JJ}^{int}$ and $\mathbf{M}_{JL}^{int}$ require integration of the product of trigonometric functions in polynomials, and the complexity of calculations grows exponentially with the number of joints.

The experimental technique is simple to implement. In the case of a real robot, it requires only the knowledge of kinematics. If the link stiffness is known, the parameters of the reduced model can be estimated from the modeling of the end-effector deflection under the applied force. The disadvantage of this approach is the dependence on the set of robot configurations and force directions. In the case of a real robot, the measurement errors affect the result as well.

The proposed technique is defined for systems with linear stiffness. In the case of nonlinearity it is better to use the generalized approach [16].

5. Conclusion

In this paper, we have considered the reduced stiffness model of the serial manipulator and demonstrated the algebraic approach for the identification of the model parameters. The

approach is based on knowledge of link compliances and robot kinematic structure. It allows finding the optimal parameter estimation for the reduced stiffness model.

The backward semianalytical Jacobian calculation method has been shown. It allows one to make the Jacobian columns from the elements of homogeneous matrices. This method was used to define the equation for the product of Jacobian columns and to simplify the proposed analytical approach.

The application of the analytical and experimental approaches is considered using a 3-link manipulator as an example. The numerical evaluation demonstrates the equivalence between these methods.

References

- [1] Alici, G. and Shirinzadeh, B., Enhanced Stiffness Modeling, Identification and Characterization for Robot Manipulators, *IEEE Trans. on Robotics*, 2005, vol. 21, no. 4, pp. 554–564.
- [2] de Luca, A. and Mattone, R., Sensorless Robot Collision Detection and Hybrid Force/Motion Control, in *Proc. of the IEEE Internat. Conf. on Robotics and Automation (Barcelona, Spain, 2005)*, pp. 999–1004.
- [3] Deblaise, D., Hernot, X., and Maurine, P., A Systematic Analytical Method for PKM Stiffness Matrix Calculation, in *Proc. of the IEEE Internat. Conf. on Robotics and Automation (Orlando, Fla., USA, 2006)*, pp. 4213–4219.
- [4] Han, Z., Yuan, J., and Gao, L., External Force Estimation Method for Robotic Manipulator Based on Double Encoders of Joints, in *ROBIO 2018: IEEE Internat. Conf. on Robotics and Biomimetics (Kuala Lumpur, Malaysia, 2018)*, pp. 1852–1857.
- [5] Hasegawa, A., Fujimoto, H., and Takahashi, T., Robot Joint Angle Control Based on Self Resonance Cancellation Using Double Encoders, in *IEEE Internat. Conf. on Advanced Intelligent Mechatronics (Munich, Germany, 2017)*, pp. 460–465.
- [6] Izumi, T. and Matsuo, T., Robot System and Robot Control Apparatus, *Patent US No. 8 849 455* (30 Sep 2014).
- [7] Klimchik, A., Furet, B., Caro, S., and Pashkevich, A., Identification of the Manipulator Stiffness Model Parameters in Industrial Environment, *Mech. Mach. Theory*, 2015, vol. 90, pp. 1–22.
- [8] Klimchik, A. and Pashkevich, A., Robotic Manipulators with Double Encoders: Accuracy Improvement Based on Advanced Stiffness Modeling and Intelligent Control, *IFAC-PapersOnLine*, 2018, vol. 51, no. 11, pp. 740–745.
- [9] Klimchik, A., Pashkevich, A., and Chablat, D., Fundamentals of Manipulator Stiffness Modeling Using Matrix Structural Analysis, *Mech. Mach. Theory*, 2019, vol. 133, pp. 365–394.
- [10] Lynch, K. and Park, F., *Modern Robotics: Mechanics, Planning, and Control*, Cambridge: Cambridge Univ. Press, 2017.
- [11] Mamedov, S., Popov, D., Mikhel, S., and Klimchik, A., Compliance Error Compensation based on Reduced Model for Industrial Robots, in *Proc. of the 15th Internat. Conf. on Informatics in Control, Automation and Robotics (ICINCO'2018): Vol. 2*, pp. 180–191.
- [12] Mamedov, S. and Mikhel, S., Practical Aspects of Model-Based Collision Detection, *Front. Robot. AI*, 2020, vol. 7, Art. 571574, 15 p.
- [13] Mikhel, S., Popov, D., Mamedov, S., and Klimchik, A., Advancement of Robots with Double Encoders for Industrial and Collaborative Applications, in *Proc. of the 23rd Conf. of FRUCT Association (Bologna, Italy, 2018)*, pp. 246–252.
- [14] Mikhel, S. and Klimchik, A., Algebraic Approach to the Stiffness Model Reduction for Manipulators with Double Encoders, in *Internat. Conf. "Nonlinearity, Information and Robotics" (Innopolis, Russia, 2020)*, pp. 1–6.
- [15] Pashkevich, A., Chablat, D., and Wenger, P., Stiffness Analysis of Overconstrained Parallel Manipulators, *Mech. Mach. Theory*, 2009, vol. 44, no. 5, pp. 966–982.

- [16] Pashkevich, A., Klimchik, A., and Chablat, D., Nonlinear Effect in Stiffness Modeling of Robotic Manipulators, *World Acad. Sci. Eng. Technol.*, 2009, vol. 58, pp. 168–173.
- [17] Piras, G., Cleghorn, W., and Mills, J., Dynamic Finite-Element Analysis of a Planar High-Speed, High-Precision Parallel Manipulator with Flexible Links, *Mech. Mach. Theory*, 2005, vol. 40, no. 7, pp. 849–862.
- [18] Popov, D. I. and Klimchik, A. S., Stiffness Modeling for Anthropomorphic Robots, *Comput. Res. Model.*, 2019, vol. 11, no. 4, pp. 631–651.
- [19] Tsai, J., Wong, E., Tao, J., McGee, H. D., and Akeel, H., Secondary Position Feedback Control of a Robot, *Patent US No. 8 473 103* (5 Aug 2013).
- [20] Sakata, K., Asaumi, H., Hirachi, K., Saiki, K., and Fujimoto, H., Self Resonance Cancellation Techniques for a Two-Mass System and Its Application to a Large-Scale Stage, *IEEE J. Ind. Appl.*, 2014, vol. 3, no. 6, pp. 455–462.
- [21] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., *Robotics: Modelling, Planning and Control*, London: Springer, 2010.
- [22] https://github.com/mikhel1984/algebraic_approach (2021).