Using extended assembly algorithm in finite element method in building dynamic equation process of flexible robot

Thuật toán lắp ghép mở rộng trong FEM nhằm xây dựng hệ phương trình động lực học của rô bốt có khâu đàn hồi

Bien Duong Xuan*, My Chu Anh

Military Technical Academy

**Email: xuanbien82@yahoo.com*

Keywords:

Assembly algorithm; Dynamic equation; Generalized vector; FEM; Flexible robot.

Abstract

Dynamic equations of flexible robot in this paper are built by using finite element method (FEM) and Lagrange's equations of the second kind approach. The generalized displacement vector in dynamic equations includes motion and elastic displacement variables. Generalized inertia matrices and stiffness matrices are established from assembling components matrices of elements. Traditional assembly method is unsuitable to assemble for generalized inertia and stiffness matrices of moving multi-body systems which have this generalized displacement vector, especially when the number of elements is incremental. Therefore, it is crucial to establish an extended general assembly algorithm for building generalized matrices based on generalized displacement vectors. This study proposed the extended general assembly algorithm which is improved based on FEM theory. This algorithm is used temporarily for single flexible link robot and two-link flexible robot with rotational or translational joints. These configures robot are also used as illustrated examples. This algorithm can serve as an useful tool for dynamics modeling of robots having flexible links with different configurations and large amount of elements.

Từ khóa:

Thuật toán lắp ghép; Hệ phương trình động lực học; Véc tơ suy rộng; Phương pháp phần tử hữu hạn; Rô bốt đàn hồi.

Tóm tắt

Trong bài báo này, phương pháp phần tử hữu hạn (FEM) và hệ phương trình Lagrange loại 2 được sử dụng để mô hình hóa động lực học cho hệ rô bốt có khâu đàn hồi. Các biến số của hệ phương trình vi phân chuyển động không chỉ có các thành phần chuyển vị đàn hồi mà còn có các thành phần biến khớp nên véc tơ biến của hệ là véc tơ biến suy rộng. Hệ số của hệ phương trình là các ma trận khối lượng và ma trận độ cứng suy rộng được hình thành từ việc lắp ghép từ các ma trận phần tử. Phương pháp lắp ghép truyền thống trở nên khó khăn khi áp dụng cho trường hợp có biến suy rộng này đặc biệt là khi số lượng phần tử tăng lên. Chính vì vậy, cần thiết phải phát triển thuật toán lắp ghép mới và mang tính tổng quát để phục vụ việc xây dựng các ma trận khối lượng và ma trận độ cứng toàn hệ thống. Bài báo này trình bày việc xây dựng thuật toán lắp ghép tổng quát cho từng khâu và cho toàn hệ thống của rô bốt đàn hồi có hai khâu nối tiếp. Mô hình rô bốt 1 khâu quay, mô hình 1 khâu tịnh tiến và mô hình rô bốt 2 khâu đàn hồi toàn khớp quay được lấy làm ví dụ minh họa. Thuật toán này có thể dùng làm công cụ rất hữu ích trong việc mô hình hóa động lực học các hệ rô bốt có khâu đàn hồi với các cấu hình khác nhau, số lượng phần tử lớn.

Received: 20/7/2018

Received in revised form: 05/9/2018

Accepted: 15/9/2018

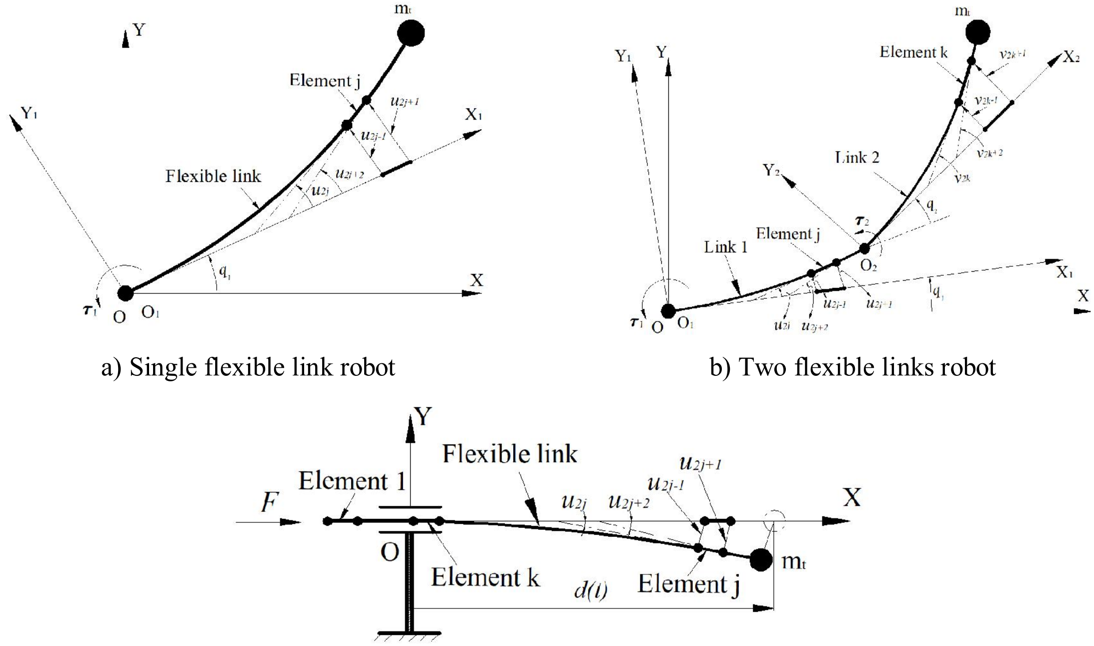
1. INTRODUCTION

In recent decades, flexible robot is very attented by researchers [1], [2], [3]. There are challengers in dynamic modeling and control because of mentioning effect of elastic displacement in motion. Lumped Parameters Method (LPM) [4], Assumed Modes Method (AMM) [5] and Finite Element Method (FEM) [6], [7], [8] are mostly used to dynamic model of flexible robot. LPM and AMM method are suitable for configures which have constant area cross-section along length of links and number of links is small normally single link or two links. Dynamic modeling problem is become complex with increasing number of links and continuous changing area cross-section. FEM is the numerical method and developed recently with the advancement of computational science. It is commonly used in mechanical structures analysis and durable calculation, analyzing dynamic behavior of system, etc. Because of development of simulation softwares, it is simple for durable and stress analyzing with FEM even through variable payload for static system. In the one hand, using FEM in dynamic modeling of motion system is complicated because of appearing of the generalized variables especially for dynamic modeling in the field of robot and flexible robot. Dynamic equations of flexible are nonlinear, many variables and complex when using FEM. Solving differential equation system is difficult, waste of time and depend on the solving method before. These problems are significantly reduced because of computational science. In the other hand, FEM is more suitable than LPM and AMM method in dynamic modeling of flexible robot with increasing number of links and continuous changing area cross-section because of dividing technique the object to multiple small elements. Dynamic equations are more simply building by combining FEM with energy Lagrange method especially configures as hybrid system (combining rigid links with flexible links, rigid joint with flexible joint, rotational joints with translational joints). Besides, using FEM in modeling is suitable for designing control system. These advantages of FEM are better than other methods.

The main problem of FEM is assembling displacement vectors, inertia and stiffness matrices of system from components vectors and matrices of elements. Assembling is simply implemented with static system because of only having elastic displacement variables but is complicated with motion system like flexible robot because of appearing generalized displacement variables. Most of flexible robot studies which used FEM are chosen each flexible link with only element or have not presented clearly assembly algorithm. Developing assembly algorithm for mechanical systems which have extra generalized variables is important meaning in modeling and building dynamic equation process of flexible robot by using FEM. This paper proposes that general assembly algorithm based on FEM theory. This algorithm is used temporarily for single flexible link robot and two-link flexible robot with rotational or translational joint. These configures robot are also used as illustrated examples. Without loss of generality, proposed assembly algorithm is presented for generalized inertia matrix. Stiffness matrix can be assembled similarly. The aim of this study is proposed assembly algorithm for generalized inertia matrix and stiffness matrix. So, dynamic modeling and building equations of motion process is not much mentioned in this study. It was clearly presented in [9], [10].

2. ILLUSTRATED FLEXIBLE ROBOTS

Considering three configures of flexible robot with rotational/translational joints and are shown in fig.1. The coordinate system XOY is the fixed frame. Coordinate system $X_1O_1Y_1$ is attached to first point of link 1. Coordinate system $X_2O_2Y_2$ is attached to first point of link 2. The rotational joints variable q_1, q_2 are driven by τ_1, τ_2 torques and translational joint is driven by F. Joints are assumed rigid. Flexible link 1 and link 2 are divided n_1, n_2 elements, respectively. The elements are assumed interconnected at certain points, known as nodes. Each element j,k, $j=1 \div n_1, k=1 \div n_2$ has two nodes. Each node of element j has 2 elastic displacement variables which are the flexural displacement (u_{2j-1}, u_{2j+1}) and the slope displacements (u_{2j}, u_{2j+2}). Similarly, node k and $k+1$ of element k have (v_{2k-1}, v_{2k}) and (v_{2k+1}, v_{2k+2}).



a) Single flexible link robot

b) Two flexible links robot

c) Single flexible link robot with translational joint

Fig. 1. Configures of illustrated flexible robot

The dynamic equation of motion relies on the Lagrange equations with Lagrange function $L=T-P$ given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{Q}}}\right)^T - \left(\frac{\partial L}{\partial \mathbf{Q}}\right)^T = \boldsymbol{\tau}(t) \tag{1}$$

where T and P are the kinetic and potential energy of the system. Vector $\boldsymbol{\tau}(t)$ is external generalized torques with rotational joints or force with translational joint acting along components of the generalized coordinate $\mathbf{Q}(t)$. Assumed that robot motions in horizontal plane, effect of gravity is can be ignored. The equations of motion can be expressed as

$$\mathbf{M}(\mathbf{Q})\ddot{\mathbf{Q}} + \mathbf{C}(\mathbf{Q},\dot{\mathbf{Q}})\dot{\mathbf{Q}} + \mathbf{D}\dot{\mathbf{Q}} + \mathbf{K}\mathbf{Q} = \boldsymbol{\tau}(t) \tag{2}$$

Where, **D** is the structural damping matrix which is can be determined in [8] and the Coriolis and centrifugal matrix is **C** which is correspondingly calculated as in [11]. The generalized inertia matrix **M** and the stiffness matrix **K** are calculated by proposed assembly algorithm based on FEM theory. All of steps building Eq. (2) can be clearly considered in [9] and [10].

3. PROPOSED GENERAL EXTENDED ASSEMBLY ALGORITHM

3.1. Assembling generalized inertia matrix of the first link

Considering the flexible link 1, $\mathbf{q}_{1j}=\left[q_1\ u_{2j-1}\ u_{2j}\ u_{2j+1}\ u_{2j+2}\right]^T$ is generalized elastic displacement vectors of elements j. Generalized inertia matrices of elements j can be described in [6] and size of that is 5×5. Generalized elastic displacement vector of link 1 is as shown below

$$\mathbf{Q}_1=\left[q_1\ u_1\ u_2\ ..\ u_{2j+1}\ u_{2j+2}\ ..\ u_{2n_1+1}\ u_{2n_1+2}\right]^T \tag{3}$$

Generalized inertia matrix of link 1 is $\mathbf{M}_1$ and calculated by assembling elements matices. The size of vector $\mathbf{Q}_1$ is $(2n_1+3)\times1$ and $\mathbf{M}_1$ is $(2n_1+3)\times(2n_1+3)$.

Firstly, considering first link with two elements ($n_1=2$), so the size of vector $\mathbf{Q}_1=\mathbf{Q}_1^{2e}$ is 7×1 and $\mathbf{M}_1$ is 7×7. We implement ticking the index for each element of $\mathbf{Q}_1^{2e}$ vector. Using these indices for $\mathbf{q}_{11}$ and $\mathbf{q}_{12}$ which are generalized vectors of element 1st and 2nd. We have

$$\mathbf{Q}_1^{2e}=[\overset{1}{q_1}\ \overset{2}{u_1}\ \overset{3}{u_2}\ \overset{4}{u_3}\ \overset{5}{u_4}\ \overset{6}{u_5}\ \overset{7}{u_6}]^T;\ \mathbf{q}_{11}=[\overset{1}{q_1}\ \overset{2}{u_1}\ \overset{3}{u_2}\ \overset{4}{u_3}\ \overset{5}{u_4}]^T;\ \mathbf{q}_{12}=[\overset{1}{q_1}\ \overset{4}{u_3}\ \overset{5}{u_4}\ \overset{6}{u_5}\ \overset{7}{u_6}]^T \tag{4}$$

The position of q_1 variable is constant in all of generalized displacement vectors of elements. So, the value of element $M_1(1,1)$ in matrix $\mathbf{M}_1=\mathbf{M}_{1\_2e}$ is sum of $M_{11}(1,1)$ and $M_{12}(1,1)$. Positions 4th and 5th in $\mathbf{q}_{11}$ are 2nd and 3rd in $\mathbf{q}_{12}$. However, their indices must be kept stable in assembly process. Values of positions which have duplicate index are adding. Note that position $(2,6),(6,2),(2,7),(7,2)$ and $(3,6),(6,3),(3,7),(7,3)$ of matrix $\mathbf{M}_{1\_2e}$ are zero because there are no indices respectively in vectors $\mathbf{q}_{11}$ and $\mathbf{q}_{12}$. Besides, generalized inertia matrix and stiffness are symmetric matrices. The matrix $\mathbf{M}_{1\_2e}$ is manually assembled and shown as below

$$\begin{array}{cc} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \mathbf{M}_{1\_2e}= & \begin{bmatrix} m_{11}^{11}+m_{11}^{12} & m_{12}^{11} & m_{13}^{11} & m_{14}^{11}+m_{12}^{12} & m_{15}^{11}+m_{13}^{12} & m_{14}^{12} & m_{15}^{12} \\ m_{21}^{11} & m_{22}^{11} & m_{23}^{11} & m_{24}^{11} & m_{25}^{11} & 0 & 0 \\ m_{31}^{11} & m_{32}^{11} & m_{33}^{11} & m_{34}^{11} & m_{35}^{11} & 0 & 0 \\ m_{41}^{11}+m_{21}^{12} & m_{42}^{11} & m_{43}^{11} & m_{44}^{11}+m_{22}^{12} & m_{45}^{11}+m_{23}^{12} & m_{24}^{12} & m_{25}^{12} \\ m_{51}^{11}+m_{31}^{12} & m_{52}^{11} & m_{53}^{11} & m_{54}^{11}+m_{32}^{12} & m_{55}^{11}+m_{33}^{12} & m_{34}^{12} & m_{35}^{12} \\ m_{41}^{12} & 0 & 0 & m_{42}^{12} & m_{43}^{12} & m_{44}^{12} & m_{45}^{12} \\ m_{51}^{12} & 0 & 0 & m_{52}^{12} & m_{53}^{12} & m_{54}^{12} & m_{55}^{12} \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \end{array} \tag{5}$$

Based on FEM theory and results of assembly above, we proposed a generally assembly algorithm for n_1 elements which is presented as below Tab. 1. (using language programing MAPLE).

3.2. Assembling generalized inertia matrix of the second link and of system

All of generalized displacement vectors on link 2 include first joint variable q_1 and elastic displacements at the end point of link $1\left(u_{2n_1+1}, u_{2n_1+2}\right)$ [6]. Generalized vectors $\mathbf{q}_{2k}$ of element k can be described as below

$$\mathbf{q}_{2k}=\begin{bmatrix} q_1 & u_{2n_1+1} & u_{2n_1+2} & q_2 & v_{2k-1} & v_{2k} & v_{2k+1} & v_{2k+2} \end{bmatrix}^T \tag{6}$$

Generalized inertia matrix of element k are shown as in [6] and size of that is 8×8. Generalized displacement vector $\mathbf{Q}_2$ of link 2 are given as [6]

$$\mathbf{Q}_2=\begin{bmatrix} q_1 & u_{2n_1+1} & u_{2n_1+2} & q_2 & v_1 & v_2 & ... & v_{2n_2+1} & v_{2n_2+2} \end{bmatrix}^T \tag{7}$$

Generalized inertia matrix of link 1 is $\mathbf{M}_2$ and calculated by assembling elements matices, respectively. The size of vector $\mathbf{Q}_2$ is $(2n_2+6)\times1$ and $\mathbf{M}_2$ is $(2n_2+6)\times(2n_2+6)$. Assumed that the second link has two elements $(n_2=2)$. Ticking the indices for elements of $\mathbf{q}_{21}, \mathbf{q}_{22}$ which are generalized vectors of elements 1st and 2nd following indices of $\mathbf{Q}_2$, we have

$$\mathbf{q}_{21}=\begin{bmatrix} \overset{1}{q_1} & \overset{2}{u_{2n_1+1}} & \overset{3}{u_{2n_1+2}} & \overset{4}{q_2} & \overset{5}{v_1} & \overset{6}{v_2} & \overset{7}{v_3} & \overset{8}{v_4} \end{bmatrix}^T \quad \mathbf{q}_{22}=\begin{bmatrix} \overset{1}{q_1} & \overset{2}{u_{2n_1+1}} & \overset{3}{u_{2n_1+2}} & \overset{4}{q_2} & \overset{7}{v_3} & \overset{8}{v_4} & \overset{9}{v_5} & \overset{10}{v_6} \end{bmatrix}^T$$
$$\mathbf{Q}_2=\begin{bmatrix} \overset{1}{q_1} & \overset{2}{u_{2n_1+1}} & \overset{3}{u_{2n_1+2}} & \overset{4}{q_2} & \overset{5}{v_1} & \overset{6}{v_2} & \overset{7}{v_3} & \overset{8}{v_4} & \overset{9}{v_5} & \overset{10}{v_6} \end{bmatrix}^T \tag{8}$$

Assembly results of generalized inertia matrix $\mathbf{M}_2=\mathbf{M}_2^{2e}$ which is implemented by the same way for first link.

$$\mathbf{M}_2^{2e}=\begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{bmatrix} m_{11}^{21}+m_{11}^{22} & m_{12}^{21}+m_{12}^{22} & m_{13}^{21}+m_{13}^{22} & m_{14}^{21}+m_{14}^{22} & m_{15}^{21} & m_{16}^{21} & m_{17}^{21}+m_{15}^{22} & m_{18}^{21}+m_{16}^{22} & m_{17}^{22} & m_{18}^{22} \\ & m_{22}^{21}+m_{22}^{22} & m_{23}^{21}+m_{23}^{22} & m_{24}^{21}+m_{24}^{22} & m_{25}^{21} & m_{26}^{21} & m_{27}^{21}+m_{25}^{22} & m_{28}^{21}+m_{26}^{22} & m_{27}^{22} & m_{28}^{22} \\ & & m_{33}^{21}+m_{33}^{22} & m_{34}^{21}+m_{34}^{22} & m_{35}^{21} & m_{36}^{21} & m_{37}^{21}+m_{35}^{22} & m_{38}^{21}+m_{36}^{22} & m_{37}^{22} & m_{38}^{22} \\ & & & m_{44}^{21}+m_{44}^{22} & m_{45}^{21} & m_{46}^{21} & m_{47}^{21}+m_{45}^{22} & m_{48}^{21}+m_{46}^{22} & m_{47}^{22} & m_{48}^{22} \\ & & & & m_{55}^{21} & m_{56}^{21} & m_{57}^{21} & m_{58}^{21} & 0 & 0 \\ & & & & m_{65}^{21} & m_{66}^{21} & m_{67}^{21} & m_{68}^{21} & 0 & 0 \\ & & & & & & m_{77}^{21}+m_{55}^{22} & m_{78}^{21} & m_{57}^{22} & m_{58}^{22} \\ & & \text{SYM} & & & & & m_{88}^{21}+m_{66}^{22} & m_{67}^{22} & m_{68}^{22} \\ & & & & & & & & m_{77}^{22} & m_{78}^{22} \\ & & & & & & & & & m_{88}^{22} \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \tag{9}$$

The proposed assembly algorithm for link 2 has a different point with algorithm which is used for first link. That is the appearing of $q_1, u_{2n_1+1}, u_{2n_1+2}$ variables in all of generalized displacement vectors on link 2. Following $\mathbf{q}_{21}, \mathbf{q}_{22}, \mathbf{Q}_2$, positions from 1st to 4th in these vectors are constant. Those positions are added by components matrix after loops, respectively. Values of positions which have duplicate index are added likely the first link. Splitting up $\mathbf{M}_{2k}$ for 4 parts (fig. 2) to assemble generalized inertia matrix $\mathbf{M}_2$. Each part is a small matrix which sizes 4×4. The algorithm for link 2nd is presented in Tab. 1.

$$\mathbf{M}_{2k}=\left[\begin{array}{c|c} \text{Part 2} & \text{Part 1} \\ \hline \text{Part 3} & \text{Part 4} \end{array}\right]$$

Fig. 2. Generalized inertia matrix of element k

Generalized displacement vector of whole system include all of joint and elastic displacement variables. It can be written as [6]

$$\mathbf{Q}=\begin{bmatrix} q_1 & u_1 & u_2 & .. & u_{2n_1+1} & u_{2n_1+2} & q_2 & v_1 & v_2 & .. & v_{2n_2+1} & v_{2n_2+2} \end{bmatrix}^T \tag{10}$$

Generalized inertia matrix of system $\mathbf{M}$ can be assembled from $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively. The size of vector $\mathbf{Q}$ is $(2n_1+2n_2+6)\times 1$ and $\mathbf{M}$ is $(2n_1+2n_2+6)\times(2n_1+2n_2+6)$. Assembling the matrix of system is simpler than other. The matrix of system is declarated with size $(2n_1+2n_2+6)\times(2n_1+2n_2+6)$ and then assembling generalized matrix of each link into this. Note that the index of positions in $\mathbf{Q}_1$ and $\mathbf{Q}_2$ vectors must be as same in $\mathbf{Q}$ vector. Assumed that $n_1=1$ and $n_2=1$, the size of $\mathbf{Q}_1$,$\mathbf{Q}_2$ and $\mathbf{Q}$ is 5×1, 8×1 and 10×1. We have

$$\begin{gathered} \mathbf{Q}_1^{1e}=\begin{bmatrix} \overset{1}{q_1} & \overset{2}{u_1} & \overset{3}{u_2} & \overset{4}{u_3} & \overset{5}{u_4} \end{bmatrix}^T;\mathbf{Q}_2^{1e}=\begin{bmatrix} \overset{1}{q_1} & \overset{4}{u_3} & \overset{5}{u_4} & \overset{6}{q_2} & \overset{7}{v_1} & \overset{8}{v_2} & \overset{9}{v_3} & \overset{10}{v_4} \end{bmatrix}^T \\ \mathbf{Q}=\begin{bmatrix} \overset{1}{q_1} & \overset{2}{u_1} & \overset{3}{u_2} & \overset{4}{u_3} & \overset{5}{u_4} & \overset{6}{q_2} & \overset{7}{v_1} & \overset{8}{v_2} & \overset{9}{v_3} & \overset{10}{v_4} \end{bmatrix}^T \end{gathered} \tag{11}$$

The matrix $\mathbf{M}$ is manually assembled and shown as below

$$\mathbf{M}_1=\begin{array}{c} \begin{array}{ccccc} 1 & 2 & . & . & 5 \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & . & . & a_{15} \\ a_{21} & a_{22} & . & . & a_{25} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{51} & a_{52} & . & . & a_{55} \end{bmatrix}_{5\times 5} \end{array} \qquad \mathbf{M}_2=\begin{array}{c} \begin{array}{ccccc} 1 & 4 & . & . & 10 \end{array} \\ \begin{bmatrix} b_{11} & b_{12} & . & . & b_{18} \\ b_{21} & b_{22} & . & . & b_{28} \\ . & . & . & . & . \\ . & . & . & . & . \\ b_{81} & b_{82} & . & . & b_{88} \end{bmatrix}_{8\times 8} \end{array}$$

$$\mathbf{M}=\begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{bmatrix} a_{11}+b_{11} & a_{12} & a_{13} & a_{14}+b_{12} & a_{15}+b_{13} & b_{14} & b_{15} & b_{16} & b_{17} & b_{18} \\ & a_{22} & a_{23} & a_{24} & a_{25} & 0 & 0 & 0 & 0 & 0 \\ & & a_{33} & a_{34} & a_{35} & 0 & 0 & 0 & 0 & 0 \\ & & & a_{44}+b_{22} & a_{45} & b_{24} & b_{25} & b_{26} & b_{27} & b_{28} \\ & & & & a_{55}+b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} \\ & & & & & b_{44} & b_{45} & b_{46} & b_{47} & b_{48} \\ & & & & & & b_{55} & b_{56} & b_{57} & b_{58} \\ & & \text{SYM} & & & & & b_{66} & b_{67} & b_{68} \\ & & & & & & & & b_{77} & b_{78} \\ & & & & & & & & & b_{88} \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \tag{12}$$

The size of $\mathbf{M}_1$,$\mathbf{M}_2$ and $\mathbf{M}$ is 5×5, 8×8 and 10×10. Applying proposed algorithms with $n_1=2$,$n_2=2$, the result is completely coincident with above result which is implemented manually.

Table 1. The extended assembly algorithm for link 1st, link 2nd and system

Algorithm for first link	Algorithm for second link	Algorithm for system

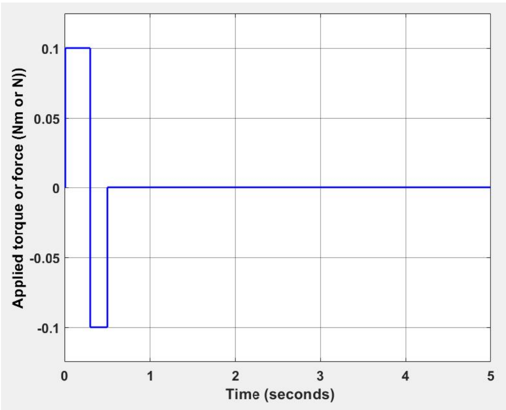
Step 1: *Declaring the size of* $\mathbf{M}_1$
$dm_1:=5+2(n_1-1);$
# Note that $(2n_1+3)=5+2(n_1-1)$ 	
$\mathbf{M}_1:=\text{Matrix}(dm_1,dm_1,0);$
$\mathbf{M}_{1j}$; # This matrix is calculated
Step 2: *Loop setup*
for j **from** 1 **to** n_1 **do**
$\mathbf{M}_{1\_e}:=\text{Matrix}(dm_1,dm_1,0);$
$\mathbf{M}_{j\_e}:=\mathbf{M}_{1j};$
# Loop setup for assembly from 2nd to 5th 	
# position in each element matrix 	
for p **from 2 to 5 do**
for q from 2 to 5 do
$M_{1\_e}[p+2(j-1),q+2(j-1)]:=M_{j\_e}[p,q];$
end;
end;
# Assembling the 1st row and 1st column 	
# except $M_1(1,1)$ position 	
for q from 2 to 5 do
$M_{1\_e}[1,q+2(j-1)]:=M_{j\_e}[1,q];$
$M_{1\_e}[q+2(j-1),1]:=M_{1\_e}[1,q+2(j-1)];$
end;
# $M_1(1,1)$ position 	
$M_{1\_e}[1,1]:=M_{j\_e}[1,1];$
# Updating total matrix $\mathbf{M}_1$ 	
$\mathbf{M}_1:=\mathbf{M}_1+\mathbf{M}_{1\_e};$
end;
	Step 1: *Declaring the size of* $\mathbf{M}_2$
	$dm_2:=8+2(n_2-1);$
	# Note that $(2n_2+6)=8+2(n_1-1)$
	$\mathbf{M}_2:=\text{Matrix}(dm_2,dm_2,0);$
	$\mathbf{M}_{2k}$; # Known
	Step 2: *Loop setup*
	for k **from** 1 **to** n_2 **do**
	$\mathbf{M}_{2\_e}:=\text{Matrix}(dm_2,dm_2,0);$
	$\mathbf{M}_{k\_e}:=\mathbf{M}_{2k};$
	# Declaring part 1
	for p **from** 1 **to** 4 **do**
	for q **from** 5 **to** 8 **do**
	$M_{2\_e}[p,q+2(k-1)]:=M_{k\_e}[p,q];$
	end;
	end;
	# Declaring part 2
	for p **from** 1 **to** 4 **do**
	for q **from** 1 **to** 4 **do**
	$M_{2\_e}[p,q]:=M_{k\_e}[p,q];$
	end;
	end;
	# Declaring part 3
	for p **from** 5 **to** 8 **do**
	for q **from** 1 **to** 4 **do**
	$M_{2\_e}[p+2(k-1),q]:=M_{k\_e}[p,q];$
	end;
	end;
	# Declaring part 4
	for p **from** 5 **to** 8 **do**
	for q **from** 5 **to** 8 **do**
	$M_{2\_e}[p+2(k-1),q+2(k-1)]=M_{k\_e}[p,q]$
	end;
	end;
	# Updating $\mathbf{M}_2$
	$\mathbf{M}_2:=\mathbf{M}_2+\mathbf{M}_{2\_e};$
	end;
		Step 1: *Initial declarating*
		$dm_1:=5+2(n_2-1);$
		$dm_2:=8+2(n_2-1);$
		$dm:=2n_1+2n_2+6$
		$\mathbf{M}_{T1}:=\text{Matrix}(dm,dm,0);$
		$\mathbf{M}_{T2}:=\text{Matrix}(dm,dm,0);$
		$\mathbf{M}:=\text{Matrix}(dm,dm,0);$
		$\mathbf{M}_1,\mathbf{M}_2$; # Known
		Step 2: Assembling $\mathbf{M}_1$ into $\mathbf{M}_{T1}$
		for i **from** 1 **to** dm_1 **do**
		for j **from** 1 **to** dm_1 **do**
		$M_{T1}[i,j]:=M_1[i,j];$
		end;
		end;
		Step 3: Assembling $\mathbf{M}_2$ into $\mathbf{M}_{T2}$
		#Assembling position $\mathbf{M}_{T2}(1,1)$
		$M_{T2}[1,1]:=M_2[1,1];$
		# Assembling 1st row and column
		# of $\mathbf{M}_2$ into $\mathbf{M}_{T2}$
		for i **from** 2 **to** dm_2 **do**
		
		$M_{T2}[1,2n_1+i]:=M_2[1,i];$
		$M_{T2}[2n_1+i,1]:=M_{T2}[1,2n_1+i];$
		end;
		# Assembling extant part of $\mathbf{M}_2$ into $\mathbf{M}_{T2}$
		for i **from** 2 **to** dm_2 **do**
		for j **from** 2 **to** dm_2 **do**
		$M_{T2}[2n_1+i,2n_1+j]:=M_2[i,j];$
		end;
		end;
		# Total matrix of system
		$\mathbf{M}:=\mathbf{M}_{T1}+\mathbf{M}_{T2};$

3. NUMERICAL SIMULATION EXAMPLES

The parameters of three configures are given in Tab.2. Applied torque and force for single flexible link robot and two-link flexible robot are shown in Fig. 2 and Fig. 3.

Table 2. Parameters of three configures flexible robot

Parameters	Single link flexible robot with rotational joint	Single link flexible robot with translational joint	Two-flexible link robot with only rotational joints
Number of elements	n_1=1;3;5;7	n_1=20	n_1=n_2=1
Length of link 1, link 2 (m)	L_1=1	L_1=1	L_1=1; L_2=0.5
Length of each element	l_e=L_1/n_1	l_e=L_1/n_1	l_{e1}=L_1; l_{e2}=L_2
Cross-section area (m^2)	A=2.5x10^{-5}	A=2.5x10^{-5}	A_1=A_2=2.5x10^{-5}
Mass of payload (kg)	m_t=0.1	m_t=0.1	m_t=0.15
Mass density (kg/m^3)	ρ=7850	ρ=7850	$\rho_1=\rho_2$=7850
Young's modulus (N/m^2)	E=2×10^{10}	E=2×10^{10}	$E_1=E_2$=2×10^{10}
Simulation time (seconds)	10	10	10



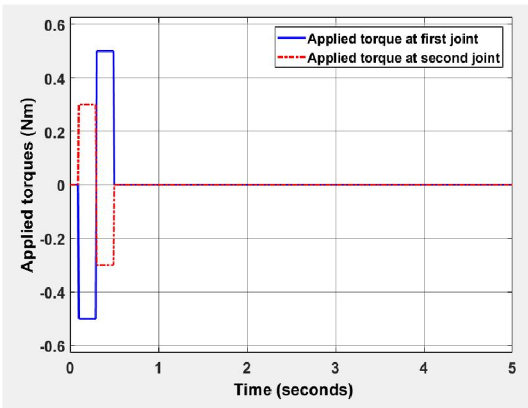


Fig. 2. Applied torque/force for single link

Fig. 3. Applied torque for two-link

The single flexible link robot with rotational joint is simulated by 4 cases: 1 element, 3 elements, 5 elements and 7 elements for flexible link. The simulated results are shown as Fig. 4. The flexible link of configure with translational joint is divided into 20 elements. The values of joint displacement and flexural displacement at the end-effector are described in Fig. 5 while simulated results of two-link flexible are presented in Fig. 6. Dividing into many elements is suitable in determining elastic displacement value at any point on flexible link.

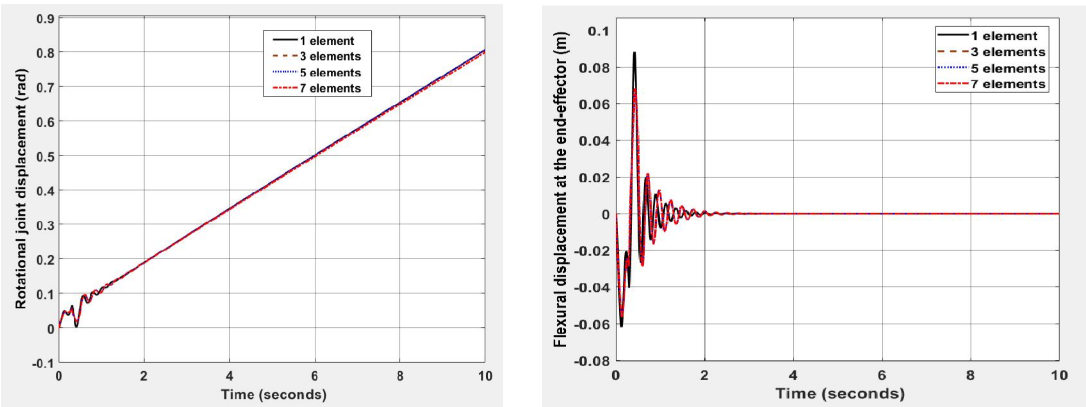


Fig. 4. Rotational joint and flexural displacement of single flexible link robot

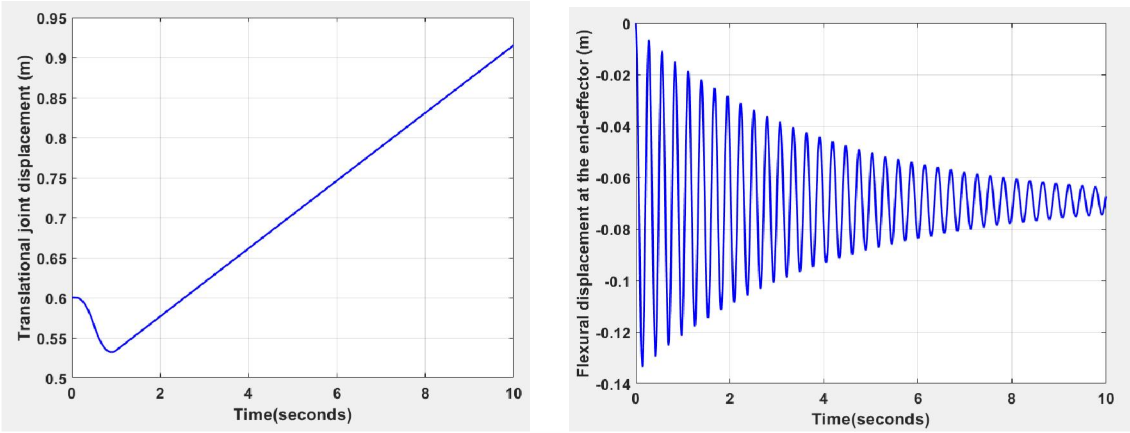


Fig. 5. Translational joint and flexural displacement of single flexible link robot

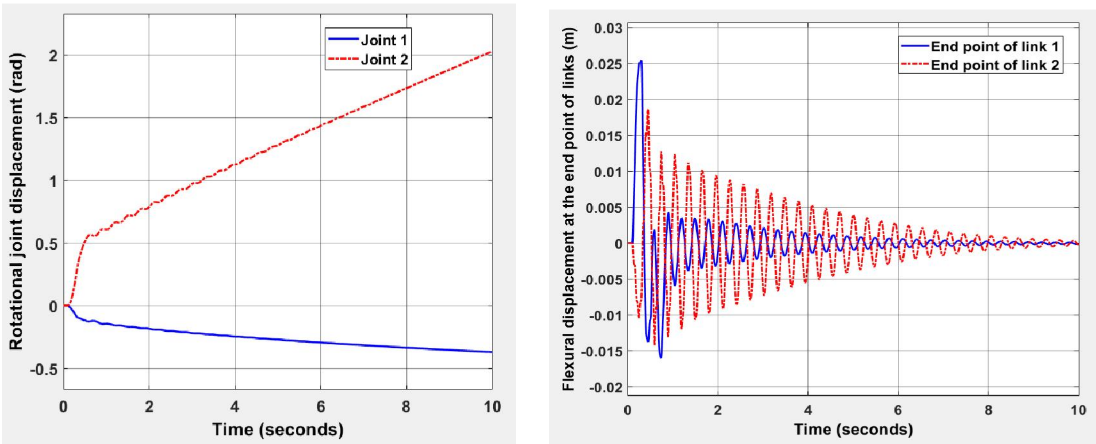


Fig. 6. Value of joints and flexural displacement of two-link flexible robot

4. CONCLUSIONS

The extended assembly algorithm in FEM is proposed and applicated for building the generalized inertia and stiffness matrices of each flexible link and system based on generalized displacement vectors. Developing assembly algorithm for mechanical systems which have extra generalized variables is important meaning in modeling and building dynamic equation process of flexible robot by using FEM. Besides, this algorithm is useful to applicate for flexible link with changing cross-section area and must be divided into many elements to analyze dynamic behavior of system. The extended algorithm is simple to implement by using MAPLE OR MATLAB language.

ACKNOWLEDMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.04-2017.09.

REFERENCES

[1]. Kiang, C. T., Spowage, A. & Yoong, C. K., 2015. Review of Control and Sensor System of Flexible Manipulator. *J. Intell. Robot. Syst. Theory Appl.* **77,** 187–213.

[2]. Lochan, K., Roy, B. K. & Subudhi, B., 2016. A review on two-link flexible manipulators. *Annu. Rev. Control* **42,** 346–367.

[3]. Sayahkarajy, M., Mohamed, Z. & Mohd Faudzi, A. A., 2016. Review of modelling and control of flexible-link manipulators. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **230,** 861–873.

[4]. Zhu, G., Lee, T. H. & Ge, S. S., 1997. Tip Tracking Control of a Single-Link Flexible Robot: A Backstepping Approach. *Dyn. Control* **7,** 341–360.

[5]. Ata, A. A., Fares, W. F. & Sa'Adeh, M. Y., 2012. Dynamic analysis of a two-link flexible manipulator subject to different sets of conditions. *Procedia Eng***,** 1253–1260.

[6]. Usoro, P. B., Nadira, R. & Mahil, S. S., 1986. A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators. *J. Dyn. Syst. Meas. Control***,** 198–205.

[7]. Al-Bedoor, B. O. & Khulief, Y. A., 1997. General Planar dynamics of a sliding flexible link. *J. Sound Vib.* **206,** 641–661.

[8]. Tokhi, M. O. & Azad, a. K. M., 2008. *Flexible Robot Manipulators*. The Institution of Engineering and Technology, London, United Kingdom.

[9]. Bien, D. X., My, C. A. & Khoi, P. B., 2017. Dynamic modeling and control in joint space of a single flexible link manipulator using Particle Swarm Optimization algorithm. *J. Sic. Tech*. 115(6), 4-8.

[10]. Bien, D. X., My, C. A. & Khoi, P. B., 2017. Dynamic analysis of two-link flexible manipulator considering the link length ratio and the payload. *Vietnam J. Mech.* **39,** 303–313.

[11]. Lewis, F. L., 2004. *Manipulator Control Theory and Practice*. Marcel Dekker, Inc, USA.

[12]. Thinh, T. I., Khoa, N. N., 2007. *Finite Element Method.* Thai Nguyen University of Technical Industry.