Development and analysis of a five degrees of freedom robotic manipulator serving as a goalkeeper to train the football players

Deep Singh\textsuperscript{1,*}, Yogesh Singh\textsuperscript{1,*}

\textsuperscript{1}Department of Mechanical Engineering, SRM Institute of Science & Technology, Kattankulathur, Chennai, Tamil Nadu, India

*Corresponding author(s): deepsingh.0318@gmail.com, yogeshsingh.a@ktr.srmuniv.ac.in

Abstract. The need of the robotic manipulators in different zones such as medical, industrial, academic and sports are available in the literatures and discussed. In the sports field like cricket, football, tennis, badminton different robotic manipulators have been used to train the players for betterment and to enhance the efficiency of the players. Football is one of the most popular game played internationally. But in the game of football we have limited invention of the robotic manipulator to train the players. Therefore, in this work the study of the development of a robotic manipulator is presented which will serve as a goalkeeper. The studied robotic manipulator has six numbers of link which is connected to the different joints and possessing five degrees of freedom. This robotic manipulator tried to defend the ball to enter the court and the simulation results and its performance have been presented here. The study will mostly be focused on the dynamics of proposed robotic manipulator using “Newton-Euler” method, “Euler-Lagrange” method and “Decoupled Natural Orthogonal Complement” (DeNOC). Due to the complexity and time consuming of Newton-Euler method and Euler-Lagrange method, the recursive Decoupled Natural Orthogonal Complement is used for the dynamic analysis and the comparison between these three methods are presented here. The effectiveness and performance of the proposed robotic manipulator with the help of different complex trajectories is presented in this work. It is found from the analysis that the proposed manipulator can be applied for the training purpose of players.

Keywords. Sports, Robotic manipulator, Tracking, Degrees of freedom, Newton-Euler method, Decoupled Natural Orthogonal Complement, Dynamics

1. Introduction

It is very essential to compute the dynamics of each and every robotic manipulator in order to understand the proper working mechanism. An ample number of serial manipulators has already been studied based on Newton-Euler and Decoupled Natural Orthogonal Complement. The study of equations of motion is
of utmost important to conclude whether the desired motion is accomplished or not. This paper is a study of branched robotic manipulator comprised of revolute as well as prismatic manipulator working in a single plane. Since it is a proposed new robotic manipulator, we have to determine the kinematics and dynamics. The derivation of equation of motion is very much important to predict the working of the proposed robotic manipulator with five degrees of freedom. The calculation is based on Newton-Euler approach. The recursive method called the Decoupled Natural Orthogonal Complement is also used to determine the equation of motion. The natural orthogonal complement concept helps in derivation of generalized inertia matrix systematically [9]. The computational simplicity of Decoupled Natural Orthogonal Complement is possible because of proper selection of robot parameters which reduces the Generalized Inertia Matrix and Matrix of Convective Inertia terms constant or zero which thereby helps in easy simplification of both the control and simulation tasks and improves its speed, stability and precision[12]. Due to the efficiency, numerical stability as well as computational uniformity of the recursive algorithms Decoupled Natural Orthogonal Complement is widely used[1][2]. Less computer memory and less processing time is required in case of Decoupled Natural Orthogonal Complement method because of smaller matrix sizes even with large systems.[8][9] To verify and understand the robotic manipulator various software has also been used to test it. The software in which the motion simulation is obtained is Adams View Student Edition[5]. For the proposed robotic manipulator, a Matlab code is also written and various forces and moments are applied to obtain the motion simulation.

2. Manipulator Design

The robotic manipulator comprises of one prismatic link (Fig.1). The prismatic link is in the upside down ‘T’ shaped. At the end point 2, two similar revolute two link manipulators each with two degrees of freedom has been assembled together to get the required manipulator. The motion of the entire robotic manipulator assembly having five degrees of freedom will be in one plane only. This makes the manipulator very simple and hence less programmable calculation making it more efficient and act within very less time.

![Figure 1. Proposed Robotic manipulator with five degrees of freedom.](image)

3. Kinematics Design

The DH parameters of the proposed robotic manipulator is given by –
Table 1. DH Parameters

| Link | Joint Offset (b<sub>i</sub>) | Joint Angle (θ<sub>i</sub>) | Link length (a<sub>i</sub>) | Twist angle (α<sub>i</sub>) |
|------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| 1    | b<sub>1</sub>               | 0                           | a<sub>1</sub>               | 90°                         |
| 2    | 0                           | θ<sub>2</sub>               | a<sub>2</sub>               | 0°                          |
| 3    | 0                           | θ<sub>3</sub>               | a<sub>3</sub>               | 0°                          |
| 4    | 0                           | θ<sub>4</sub>               | a<sub>4</sub>               | 0°                          |
| 5    | 0                           | θ<sub>5</sub>               | a<sub>5</sub>               | 0°                          |

The Homogeneous Transformation matrix is written as mentioned below [6].

\[
T_i = \begin{bmatrix}
\cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\
\sin \theta_i & \cos \theta_i \sin \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\
0 & \sin \alpha_i & \cos \alpha_i & b_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

If we divide the robotic manipulator into two manipulators then we can see that these two robotic manipulators are in the form of –

i) Prismatic – Revolute – Revolute (PRR) or (1-2-3)

ii) Revolute – Revolute (4-5)

From the fig. we can easily say that 2-3 and 4-5 are different manipulators but with same configuration (Revolute-Revolute) possessing same characteristics except the magnitude values. So, just finding the kinematics of link 2-3 or 1-2-3 robotic manipulator serves the purpose after which both of it can be combined together using the Lagrange multiplier.

So, the homogeneous transformation matrix for the robotic manipulator 1-2-3 is obtained to be:

\[
T = T_1 T_2 T_3 = \begin{bmatrix}
\cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_1 \\
0 & 0 & -1 & 0 \\
\sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & a_3 \sin(\theta_2 + \theta_3) + a_2 \sin(\theta_2) + b_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

4. Dynamics Analysis

4.1. Jacobian matrix

If we consider the links 2-3 and 4-5, we can say that both are two link robotic manipulators which has been combined together. So, the Jacobian matrix and all the various calculation for both the links 2-3 and 4-5 will be same.

Let us now represent the links with 2-3 or 4-5 as any different two link manipulator whose links are 1-2. Therefore, the figure of the robotic manipulator with two revolute links having two degrees of freedom obtained is shown below:
Figure 2. Two link revolute robotic manipulator

Jacobian, \( J = [e_1 \times a_{1e} \quad e_2 \times a_{2e}] \)

\[ e_1 = e_2 = [0 \ 0 \ 1]^T \]

\[ a_{1e} = a_1 + a_2 = a_1c_i + a_1s_i + a_2c_{12} + a_2s_{12} = \begin{bmatrix} a_1c_i + a_2c_{12} \\ a_1s_i + a_2s_{12} \\ 0 \end{bmatrix} \]

\[ a_{2e} = a_2c_{12} + a_2s_{12} = \begin{bmatrix} a_2c_{12} \\ a_2s_{12} \\ 0 \end{bmatrix} \]

\[ J = \begin{bmatrix} -a_1s_i - a_2s_{12} & -a_2s_{12} \\ a_1c_i + a_2c_{12} & a_2c_{12} \\ 0 & 0 \end{bmatrix} \]

4.2. Newton Euler Equation for two link revolute arm

Inverse and forward dynamics can be used to generate the equations of motion [7].

4.2.1. Forward Computation.

Rotation Matrices,

\[ Q_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

Link length,

\[ [a_1]_i = [a_1c_i \quad a_1s_i \quad 0]^T, \quad [a_2]_i = [a_2c_2 \quad a_2s_2 \quad 0]^T \]

\[ [d_1]_i = \left[ \frac{1}{2}a_2c_i \quad \frac{1}{2}a_1s_i \quad 0 \right]^T, \quad [d_2]_i = \left[ \frac{1}{2}a_2c_2 \quad \frac{1}{2}a_2s_2 \quad 0 \right]^T \]

\[ [r]_{i+1} = \left[ \frac{a_i}{2} \quad 0 \quad 0 \right]^T \]

\[ [w]_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_i \end{bmatrix}, \quad [\dot{w}]_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_i \end{bmatrix} \]
\[
[c_1]'_1 = \dot{\theta}_1 \frac{a_1}{2} \left[ -\frac{s_1}{c_1} \right], \quad [c_1]'_1 = \ddot{\theta}_1 \frac{a_1}{2} \left[ -\frac{s_1}{c_1} \right] - \dot{\theta}_1^2 \frac{a_1}{2} \left[ \frac{s_1}{c_1} \right] \\
[l_1]'_2 = \frac{m_1 a_2}{12} \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad [l_2]'_3 = \frac{m_3 a_2^2}{12} \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] \\
g'_1 = \left[ \begin{array}{c} 0 \\ -g \\ 0 \end{array} \right]^T, \quad [g]_2 = Q_1[g]'_1 = \left[ -g s_1 \right] \\
[w_2]'_2 = \left[ \begin{array}{c} 0 \\ 0 \end{array} \right], \quad [w_2]'_2 = \left[ \begin{array}{c} 0 \\ \dot{\theta}_{12} \end{array} \right] \\
[c_2]'_2 = Q_1'([c_1]'_1 + [w_1]'_1 \times [r_1]'_1) + [w_2]'_2 \times [d_2]'_2 = \left[ \begin{array}{c} -\frac{a_2}{2} s_2 \dot{\theta}_{12} \\ \frac{a_2}{2} \dot{\theta}_{12} \\ 0 \end{array} \right] \\
[c_2]'_2 = \left[ \begin{array}{c} -a_1 \dot{\theta}_1^2 - \frac{a_2}{2} \left( s_2 \dot{\theta}_{12} + c_2 \dot{\theta}_{12} \right) \\ a_1 \dot{\theta}_1^2 + \frac{a_2}{2} \left( c_2 \dot{\theta}_{12} - s_2 \dot{\theta}_{12} \right) \\ 0 \end{array} \right] \\
4.2.2. Backward Computation

Assumption: No external force has been applied.

Therefore, \([f_2]'_3 = [n_2]'_3 = 0, \quad [f_2]'_3 = [n_2]'_3 = 0\)

Since \(f_{23} = f_{12}, \quad n_{23} = n_{32}\)

Now, \([f_2]'_2 = m_2 [c_2]'_2 = m_2 \left[ \begin{array}{c} -a_1 \dot{\theta}_1^2 - \frac{a_2}{2} \left( s_2 \dot{\theta}_{12} + c_2 \dot{\theta}_{12} \right) \\ a_1 \dot{\theta}_1^2 + \frac{a_2}{2} \left( c_2 \dot{\theta}_{12} - s_2 \dot{\theta}_{12} \right) \\ 0 \end{array} \right] \)

\([n_2]'_2 = [l_2]'_2 [w_2]'_2 = \frac{m_2 a_2^2}{12} \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \)

\([f_2]'_2 = [f_2]'_2 + [f_2]'_3 + m_2 [g]'_2 = m_2 \left[ \begin{array}{c} -a_1 \dot{\theta}_1^2 - \frac{a_2}{2} \left( s_2 \dot{\theta}_{12} + c_2 \dot{\theta}_{12} \right) + g s_1 \\ a_1 \dot{\theta}_1^2 + \frac{a_2}{2} \left( c_2 \dot{\theta}_{12} - s_2 \dot{\theta}_{12} \right) + g c_1 \\ 0 \end{array} \right] \]

\(= \left[ \begin{array}{c} (f_{12})_x \\ (f_{12})_y \end{array} \right] \) (let)
\[ [n_{12}]_2 = [n_2]_2 + [n_{23}]_2 + [d_2]_2 \times [f_{12}]_2 + [r_1]_1 \times [f_{23}]_2 \]
\[ = \begin{bmatrix} \frac{a_2}{2} s_2 (f_{12})_x + \frac{a_2}{2} c_2 (f_{12})_y + \frac{m_2 a_2^2}{12} \theta_{12} \gamma & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \]

\[ [f_{11}]_1 = m_1 [c_1]_1 = m_1 \frac{a_1}{2} \left( \theta^1 \begin{bmatrix} -s_1 \\ 0 \\ c_1 \end{bmatrix} - \theta^2 \begin{bmatrix} 0 \\ c_1 \\ s_1 \end{bmatrix} \right) \]

\[ [n_1]_1 = [I_1]_1 \begin{bmatrix} \dot{\theta}^1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{m_1 a_1^2}{12} \dot{\theta}^1 \end{bmatrix} \]

\[ [f_{01}]_1 = [f_{11}]_1 + [f_{12}]_1 - m_1 [g]_1 \]
\[ = \begin{bmatrix} c_1(f_{12})_x - s_1(f_{12})_y - \frac{m_1 a_1}{2} \left( s_1 \dot{\theta}^1 + c_1 \theta^2 \right) \\ s_1(f_{12})_x + c_1(f_{12})_y + \frac{m_1 a_1}{2} \left( c_1 \dot{\theta}^1 - s_1 \theta^2 \right) + m_1 g \end{bmatrix} \]

\[ [n_{01}]_1 = [n_1]_1 + [n_{12}]_1 + [d_1]_1 \times [f_{01}]_1 + [r_1]_1 \times [f_{12}]_1 \]
\[ = \begin{bmatrix} 0 \\ \frac{m_1 a_1^2}{12} \dot{\theta}^1 + (n_{12})_2 + a_1(f_{12})_y + m_1 g \frac{a_1}{2} c_1 \end{bmatrix} \]

\[ 4.2.3. \text{Joint Torque.} \]
\[ \tau_1 = [e_1]^T [n_{01}]_1 = \frac{m_1 a_1^2}{3} \ddot{\theta}^1 + \frac{1}{2} m_1 a_1 a_2 s_2 \ddot{\theta}^2 + \frac{m_2 a_2^2}{3} \theta_{12} + m_2 g \left( \frac{a_2}{2} c_1 + a_1 c_1 \right) + m_1 g \frac{a_1}{2} c_1 
+ \frac{1}{2} m_2 a_1 a_2 c_2 \ddot{\theta}_1 + m_2 a_1 \left( a_1 \ddot{\theta}_1 + \frac{a_2}{2} (c_2 \ddot{\theta}_{12} - s_2 \theta_{12}) \right) \]
\[ \tau_2 = [e_2]^T [n_{12}]_2 = \frac{m_2 a_2^2}{3} \ddot{\theta}_{12} + \frac{1}{2} m_2 a_1 a_2 \left( c_2 \ddot{\theta}_1 + s_2 \ddot{\theta}^2 \right) + m_2 g \frac{a_2}{2} c_1 \]

\[ 4.3. \text{Recursive Dynamics (DeNOC)} \]
\[ \dot{\theta} = \begin{bmatrix} \dot{\theta}^1 \\ \dot{\theta}^2 \end{bmatrix} \]

\[ 4.3.1. \text{Generalized Inertia Matrix (I). The expression for Generalized Inertia Matrix is mentioned below [3][4].} \]
\[ I = \begin{bmatrix} i_{11} & i_{12} \\ i_{21} & i_{22} \end{bmatrix} \]
\[ i_{22} = \bar{P}_2^T \bar{M}_2 \bar{B}_{22} \bar{P}_2 \]
\[ i_{22} \]

6
Here, \[ p_2 = \left[ \begin{array}{c} e_2 \\ e_2 \times d_2 \end{array} \right], \quad b_{22} = \left[ \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \quad \widehat{M}_2 = \left[ \begin{array}{cc} I_2 & 0 \\ 0 & m_2 1 \end{array} \right] \]

Therefore, \[ i_{22} = \frac{1}{3} m_2 a_2^2 \]
\[ i_{21} = p_2^T \widehat{M}_2 b_{21} p_1 \]

Here, \[ b_{21} = \left[ \begin{array}{cc} \left( r_1 + d_2 \right) \times 1 & 0 \end{array} \right], \quad p_1 = \left[ \begin{array}{c} e_1 \\ e_1 \times d_1 \end{array} \right] \]

Therefore, \[ i_{21} = \frac{1}{3} m_2 a_2^2 + \frac{1}{2} m_2 a_1 a_2 c_2 \]

Similarly, \[ i_{11} = e_1^T \overline{r}_1 e_1 + \overline{m}_1 d_1^T d_1 \]

Here, \[ \overline{m}_1 = m_1 + m_2 \]

Therefore, \[ i_{11} = \frac{1}{3} \left( m_1 a_1^2 + m_2 a_2^2 \right) + m_2 a_1^2 + m_2 a_1 a_2 c_2 \]

### 4.3.2. Matrix for Convective Inertia (C). The expression for Matrix for Convective Inertia is mentioned below\(^{[3]}\)\(^{[4]}\).

\[ c_{11} = 0 \]
\[ c_{12} = -\frac{1}{2} m_2 a_1 a_2 s_2 \dot{\theta}_{12} \]
\[ c_{21} = \frac{1}{2} m_2 a_1 a_2 s_2 \dot{\theta}_1 \]
\[ c_{21} = -\frac{1}{2} m_2 a_1 a_2 s_2 \dot{\theta}_2 \]

### 4.3.3. Equations of motion.

The Equation of motion is given as, \[ \tau = I \ddot{\theta} + C \dot{\theta} \]

Hence, \[ h_2 = p_2^T \widetilde{w}_2 = \frac{1}{2} m_2 a_1 a_2 s_2 \dot{\theta}_1^2 \]
\[ h_1 = p_1^T \widetilde{w}_1 = -m_2 a_1 a_2 s_2 \left( \frac{1}{2} \dot{\theta}_2 + \dot{\theta}_1 \right) \dot{\theta}_2 \]

### 4.4. Lagrange Multiplier

The kinematics and dynamics are derived for the two links revolute robotic manipulators. A lagrange multiplier (\(\lambda\)) is used to combine two different two link revolute robotic manipulators of same specifications to get the desired robotic manipulator \(^{[1]}\)\(^{11}\). The other prismatic link will be placed on the ground.

So, the two different two link revolute manipulators can be named as ‘i’th manipulator and ‘j’th manipulator and the various operations as well as mathematical calculations will be carried appropriately.

For system ‘\(i\)’

Unconstrained Newton Euler equations is represented as:
\[ M \ddot{t} + C t = w \]

where \[ M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \quad w = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} \]

\( t = \) twist \quad \dot{t} = \) twist rate

Converting the above into unconstrained equation can be given as,

\[ N^T w^* = \tau^e + \tau^\lambda \quad \text{where} \quad \tau^e = N^T w^e, \quad \tau^\lambda = N^T w^\lambda \]

\[ N = \begin{bmatrix} P_1 \\ A_{21} P_1 \\ P_2 \end{bmatrix}, \quad A_{21} = \begin{bmatrix} 1 \\ E a_1 \\ 0 \end{bmatrix} \]

where \( \tau^e \) and \( \tau^\lambda \) are external torque and torque due to lagrange multiplier.

And, the external forces and moments results in wrenches such as

\[ w_1^e = \begin{bmatrix} 1 \\ 0 \\ \tau^e \end{bmatrix}, w_2^e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

Hence the constrained equations can be obtained as,

\[ (a_1 + a_2) \bar{E} \dot{\lambda} - \tau_\lambda = n_1 + n_2 - a_1^T \bar{E} f_2^e - n_1^e + n_2^e - n_2^e + b_1^T \bar{E} f_1^e + b_2^T \bar{E} f_2^e + a_1^T \bar{E} f_2^e \]

And \( a_2^T \bar{E} \dot{\lambda} = n_2^e - b_2^T \bar{E} f_2^e \)

For system ‘i’

By following the above similar procedure we can obtain the constrained equations for the jth system also and both together can be used to understand the dynamics and motions of the end effector of the robotic manipulator that has been proposed.

5. Simulation

5.1. Geometrical and Material Parameters used for simulation

The various geometrical and parameters which has been used for the modelling the simulation are mention below :

| Material | Density (kg/mm³) | Young’s Modulus (Newton/mm²) | Poisson’s Ratio | Length : a₁, a₂ |
|----------|-----------------|-----------------------------|----------------|-----------------|
| Steel    | 7.801 × 10⁻⁶    | 2.07 × 10⁵                  | 0.29           | 1m, 0.5m        |

5.2. Based on trajectory motion

The simulation has been carried out in the ‘ADAMS View’ software and verified the result using the ‘MATLAB’ software for various end-effector trajectory out of which three has been detailed below :

i) Trajectory 1
Displacement along X axis : $10\cos(0.1\times\text{time})$

Displacement along Y axis : $10\sin(0.1\times\text{time})$

Duration : 100, Step size : 1

Figure 3. ADAMS View Graph

Figure 4. MATLAB Graph

ii) Trajectory 2

Displacement along X axis : $10\cos(0.1\times\text{time})$

Duration : 100, Step size : 1
iii) Trajectory 3

Displacement along Y axis : -5*sin(0.1*time)

Duration : 100, Step size : 1
5.3. Based on force at the end effector

The simulation has been carried out in the ‘ADAMS View’ software and verified the result using the ‘MATLAB’ software for various forces introduced at the end-effector out of which two has been detailed below:

i) Trajectory 1

Force along X axis : 0.01N

Duration : 50,     Step size : 1
ii) Trajectory 2

Force along X axis : 0.01N
Force along Y axis : 0.01N
Duration : 50, Step size : 1
6. Conclusion

The dynamics has been obtained using the ADAMS View software for the proposed robotic manipulator and verified using the MATLAB software for various trajectories. So, it is possible to design and fabricate such proposed robotic manipulators with five degrees of freedom. Hence, this type of robotic manipulator can be designed and fabricated for the purpose of training of football players. The proposed robotic manipulator can be used as a goalkeeper for the enhancement of efficiency of football players.

7. Nomenclature

| Parameter | Nomenclature |
|-----------|--------------|
| $\dot{\theta}$, $\ddot{\theta}$ | The n-dimensional vectors of generalized rates and accelerations respectively. |
| $a_i$ | The 3-dimensional vector denoting point $O_{i+1}$ from $O_i$. |
| $B_{ij}$ | The 6x6 twist propagation matrix. |
| $\dot{\mathbf{c}}_i$, $\ddot{\mathbf{c}}_i$ | The 3-dimensional linear velocity and acceleration vectors of the mass center of link $i$ respectively. |
| $M$, $\bar{M}$ | The 6n x 6n generalized and composite mass matrix respectively. |
| $P_i$ | The 6-dimensional joint motion propagation vector. |
| $p$ | Position vector of any point. |
| $t_e$ | The 6-dimensional twist vector of the end-effector. |
| $t_i$ | The 6-dimensional twist vector associated with the $i^{\text{th}}$ body. |
| $\dot{t}_i$ | The 6-dimensional twist-rate vector of the $i^{\text{th}}$ body. |
| $1$ | Identity matrix. |
| $O$ | Zero matrix of compatible dimension. |
| $0$ | Zero vector of compatible dimension. |
| $w$ | 6n dimensional generalized vector. |
| $w_i$ | 6 dimensional wrench vector acting on the $i^{\text{th}}$ body. |
8. References

[1] Raoofian A, Ali Kamali E and Taghvaeipour A 2017 Forward dynamic analysis of parallel robots using modified decoupled natural orthogonal complement method, *Mechanism and Machine Theory*. 115, 197-217.

[2] Saha S K, Shah S V and Nandihal P V 2013 Evolution of the DeNOC-based dynamic modelling for multibody systems, *Mech. Sci.*, 4, 1-20.

[3] Saha S K 1999 Dynamics of Serial Multibody Systems Using the Decoupled Natural Orthogonal Complement Matrices, *ASME. J. Appl. Mech.*, 66(4), 986-996.

[4] Khan W A, Krovi V N, Saha S K and Angeles J 2004 Recursive Kinematics and Inverse Dynamics for a Planar 3R Parallel Manipulator. *ASME. J. Dyn. Sys., Meas., Control*, 127(4), 529-536.

[5] Huang H, Dong E, Xu M, Yang J and Low K H 2018 Mechanism design and kinematic analysis of a robotic manipulator driven by joints with two degrees of freedom (DOF), *Industrial Robot: An International Journal*. 45(1), 34-43.

[6] Shah S V, Saha S K and Dutt J K 2012 Denavit-Hartenberg Parameterization of Euler Angles, *ASME. J. Comput. Nonlinear Dynam.*, 7(2).

[7] Saha S K 1997 A decomposition of the manipulator inertia matrix. *IEEE Transactions on Robotics and Automation*. 13(2), 301-304.

[8] Hanzaki A R, Saha S K and Rao 2009 Dynamic Modeling of Multibody Systems with Spherical Joints using Euler Parameters. *Multibody System Dynamics*, 21(4), 325-345.

[9] Angeles J and Ma O 1988 Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement. *The International Journal of Robotics Research*, 7(5), 32-47.

[10] Saha S K and Schiehlen W O 2001 Recursive Kinematics and Dynamics for Parallel Structured Closed-loop Multibody Systems, *Mechanics of Structures and Machines, An International Journal*. 29(2), 143-175.

[11] Chaudhary H and Saha S K 2009 Dynamics of Closed-loop Systems, *In: Dynamics and Balancing of Multibody Systems. Lecture Notes in Applied and Computational Mechanics*, Springer, Berlin, Heidelberg, 37.

[12] Bhangale P P, Saha S K and Agarwal V P 2003 Robot Selection using DeNOC-based dynamics. *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, 4, 1544-1548.