Dynamic Modelling of Mixed Rigid-Flexible Joint Robotic Manipulator Using Recursive Newton-Euler Formulation

Minh Tuan Hua
Department of Engineering Sciences
University of Agder
Grimstad, Norway
minh.tuan.hua@uia.no

Mohammad Poursina
Department of Engineering Sciences
University of Agder
Grimstad, Norway
poursina@uia.no

Filippo Sanfilippo
Department of Engineering Sciences
University of Agder
Grimstad, Norway
filippo.sanfilippo@uia.no

***Abstract*—Rigid joints have good position control accuracy due to their stiffness. On the other hand, elastic joints are advantageous when interacting with the environment due to their compliance characteristics. Most of the current robot arms use all rigid joints or all elastic joints in their designs. This work presents a highly efficient approach to formulate both inverse and forward dynamics of robots with mixed rigid-elastic joints using recursive Newton-Euler algorithm. To simplify the modelling process, a unified rigid body is proposed, where the link and all motors attached to it are unified as a whole unyielding entity. In addition, the effect of gear ratio is considered in the development of this modelling approach. Successively, an inverse dynamics controller is presented. Finally, simulations are conducted based on the proposed modelling method and the inverse dynamics control algorithm.**

***Index Terms*—recursive Newton-Euler algorithm, modelling, inverse dynamics control, flexible joint manipulator**

I. Introduction

Typical industrial robot manipulators have rigid joints, hence they can render high torque and have outstanding accuracy in trajectory tracking control. However, elastic joints are more compatible when interacting with the environment, including humans, because of their safety and shock-absorbing ability. Nevertheless, elastic joints can negatively affect position accuracy and response time compared to the rigid ones. To harmonise the benefits and alleviate the drawbacks of such joints, robots with mixed rigid-elastic joints can be introduced.

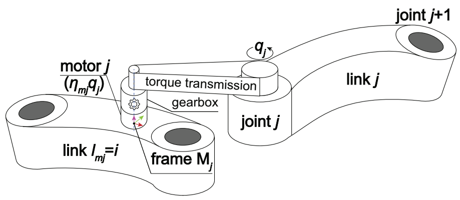
The predominant focus in existing research endeavours concerning elastic joints pertains to the exclusive modelling and control methodologies applied to robots equipped solely with elastic joints. In [1], a recursive Newton-Euler algorithm is proposed to calculate the inverse dynamics of a serial robotic manipulator with elastic joints. However, this algorithm requires the fourth-order derivative of the joint angles. In [2], an adaptive fuzzy global coupled nonsingular fast terminal sliding mode control is proposed to control robots with elastic joints in the presence of uncertainties. In [3], a two-feedback loops position controller is presented. The inner loop is a model reference adaptive controller for controlling motors and the outer loop is a fuzzy proportional-integral controller generating desired positions for the motors. In [4], the two-feedback loops position controller was tested on a real mechanical model of a two degrees of freedom (DOFs) robot with elastic actuators.

To the best of our knowledge, there is a scarcity of literature addressing robotic arms that incorporate both rigid and elastic joints. In [5], static state feedback and dynamic state feedback control algorithms are proposed. The dynamic state feedback introduces a linear dynamic feedback compensator for the inputs at the rigid joints, which softens the rigid joints. In [6], an adaptive control algorithm for controlling a mixed rigid-flexible joint manipulator based on the virtual decomposition method is proposed.

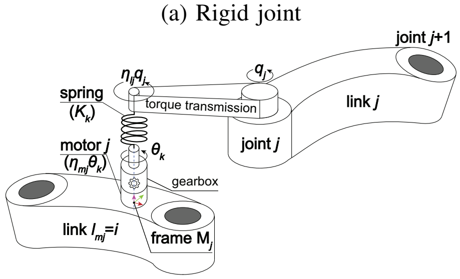
This paper proposes recursive Newton-Euler algorithm to compute the inverse dynamics of robots with mixed rigid-elastic joints. Moreover, the forward dynamics algorithm to determine joint accelerations, given current joint positions and velocities, and motor torques, is presented. In addition, the motor position assumption is removed. Commonly, it is assumed that the motors are placed preceding the driven link. In the approach presented in this paper, the motors are assumed to be placed on any links on the robotic manipulator. This is a reasonable assumption because, in some industrial robots, the motors are placed as close as possible to the base to lower the centre of mass of the robot and reduce the load of the links close to the base. Additionally, a novel approach is adopted through the utilisation of unified rigid bodies, wherein a link and all motors attached to it are considered as a single firm entity. Then, an inverse dynamics controller is proposed to control the robotic arm. Ultimately, simulation experiments are conducted utilising the dynamic model to mimic a three-degree-of-freedom (DOFs) robot equipped with a combination of rigid and elastic joints.

The paper is organised as follows. In Section II, the recur-

This research is supported by the Biomechatronics and Collaborative Robotics research group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.



(a) Rigid joint



(b) Elastic joint

Fig. 1: The kinematic diagram of a rigid and an elastic joint.

sive Newton-Euler algorithm to compute the inverse dynamics is presented. Then, in Section III, the algorithm for computing the forward dynamics using the recursive Newton-Euler algorithm is proposed. In Section IV, the inverse dynamics controller is outlined. In Section V, simulation results are carried out. Finally, in Section VI, conclusion are drawn.

II. Modelling

In this work, the following assumptions are considered:

A1: the elasticity region is limited in the linear elasticity domain, where the Hooke's law is applicable.

A2: the rotors of the motors are modelled as uniform bodies with their centres of mass on the rotation axis.

A robotic manipulator can be described as an open kinematic chain where $N+1$ rigid bodies, encompassing the base link 0 and N additional links, are interconnected by N joints and driven by N electrical motors. Link $i-1$ and link i are connected via joint i. Assume that there are $N_e < N$ elastic joints and $N_r = N - N_e$ rigid joints. Let l_{mi} denote the link where the motor i is located. Figures 1a and 1b, respectively, illustrate a rigid and an elastic joint. In these figures, the driving motors are illustrated to be attached to link $l_{mj} = i$. Let q_i and θ_k, respectively, denote the angular position of joint i and the angular position of the motor driving the k^{th} elastic joint. Notice that θ_k is not the angular position of the motor controlling joint k, but the angular position of the motor of the k^{th} elastic joint. In this work, link i and all motors attached to it will be considered as a unified rigid body i, as illustrated in Figure 2. A coordinate system (frame i) is assigned for each joint i, where the axis z_i is located along the rotational axis. In addition, a coordinate system (frame M_i) is defined for each motor i, while its z-axis (denoted as $\mathbf{z}_{mi}^0$ when expressed in the base frame 0) is located along the motor's rotation axis. Therefore, based on the assumption A2, the inertia tensor of motor i expressed in frame M_i, denoted as $\boldsymbol{I}_{mi}^{M_i}$, is a diagonal constant matrix

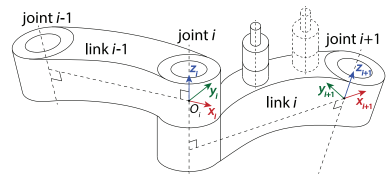


Fig. 2: Unified rigid body i, including link i and all motors attached to it.

defined as $\boldsymbol{I}_{mi}^{M_i}$=$diag(I_{mxi}, I_{myi}, I_{mzi})$. Furthermore, vector $\mathbf{z}_{mi}^0$ when expressed in frame M_i becomes $\mathbf{z}_{mi}^{M_i} = [0\ 0\ 1]^T$. Moreover:

$$\boldsymbol{I}_{mi}^0\mathbf{z}_{mi}^0 = \boldsymbol{R}_{M_i}^0\boldsymbol{I}_{mi}^{M_i}(\boldsymbol{R}_{M_i}^0)^T\mathbf{z}_{mi}^0 = \boldsymbol{R}_{M_i}^0 I_{mzi}\mathbf{z}_{mi}^{M_i} = I_{mzi}\mathbf{z}_{mi}^0 \quad (1)$$

where $\boldsymbol{R}_i^0$ denotes the rotation matrix of frame i with respect to the base frame.

A. Forward Kinematic

In the forward kinematic procedure, velocities and accelerations are recursively propagated from the base link to the end-effector. Let $\boldsymbol{\omega}_i^0$ and $\boldsymbol{\omega}_{mj}^0$, respectively, denote the angular velocity of link i and motor j represented in frame 0. The angular velocity of link i is the sum of the angular velocity of link $i-1$ and the angular velocity of joint i:

$$\boldsymbol{\omega}_i^0 = \boldsymbol{\omega}_{i-1}^0 + \dot{q}_i\mathbf{z}_i^0 = \boldsymbol{\omega}_{i-1}^0 + \dot{q}_i\boldsymbol{R}_i^0\boldsymbol{z}_0 \quad (2)$$

where $\mathbf{z}_0 = [0\ 0\ 1]^T$, and $\mathbf{z}_i^i = \mathbf{z}_0$.

Taking the time derivative of $\boldsymbol{\omega}_i^0$ to obtain the angular acceleration of link i results in:

$$\dot{\boldsymbol{\omega}}_i^0 = \dot{\boldsymbol{\omega}}_{i-1}^0 + \ddot{q}_i\mathbf{z}_i^0 + \dot{q}_i\dot{\boldsymbol{R}}_i^0\mathbf{z}_0 = \dot{\boldsymbol{\omega}}_{i-1}^0 + \ddot{q}_i\mathbf{z}_i^0 + \dot{q}_i\overset{\times}{\boldsymbol{\omega}}{}_{i-1}^0\mathbf{z}_i^0 \quad (3)$$

where $\overset{\times}{\mathbf{a}}$ denote the skew-symmetric matrix of vector $\mathbf{a} = [a_1\ a_2\ a_3]^T$, and $\overset{\times}{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$.

To reduce the computational burden, vectors $\boldsymbol{\omega}_i^0$ and $\dot{\boldsymbol{\omega}}_i^0$ are expressed in frame $i+1$ rigidly attached to frame i by multiplying with the rotation matrix $(\boldsymbol{R}_{i+1}^0)^T$:

$$\boldsymbol{\omega}_i^{i+1} = (\boldsymbol{R}_{i+1}^0)^T\boldsymbol{\omega}_i^0 = (\boldsymbol{R}_{i+1}^i)^T(\boldsymbol{\omega}_{i-1}^i + \dot{q}_i\mathbf{z}_0) \quad (4)$$

$$\dot{\boldsymbol{\omega}}_i^{i+1} = (\boldsymbol{R}_{i+1}^0)^T\dot{\boldsymbol{\omega}}_i^0 = (\boldsymbol{R}_{i+1}^i)^T(\dot{\boldsymbol{\omega}}_{i-1}^i + \ddot{q}_i\mathbf{z}_0 + \dot{q}_i\overset{\times}{\boldsymbol{\omega}}{}_{i-1}^i\mathbf{z}_0) \quad (5)$$

Let η_{mj} denote the gear ratio of motor j. The angular velocities and accelerations of the motors are derived according to the type of joints they drive. Referring to Figure 1b, if joint j is the k^{th} elastic joint, and motor j is attached to link i (e.i. $l_{mj} = i$), then:

$$\boldsymbol{\omega}_{mj}^0 = \boldsymbol{\omega}_i^0 + \eta_{mj}\dot{\theta}_k\mathbf{z}_{mj}^0 = \boldsymbol{\omega}_i^0 + \eta_{mj}\dot{\theta}_k\boldsymbol{R}_{i+1}^0\mathbf{z}_{mj}^{i+1} \quad (6)$$

Notice that the vector $\mathbf{z}_{mj}^{i+1}$ is a constant vector because both motor j and frame $i+1$ are rigidly attached to link i. Taking the time derivative of $\boldsymbol{\omega}_{mj}^0$, the angular acceleration of the motor j is expressed as:

$$\begin{aligned} \dot{\boldsymbol{\omega}}_{mj}^0 &= \dot{\boldsymbol{\omega}}_i^0 + \eta_{mj}\ddot{\theta}_k\mathbf{z}_{mj}^0 + \eta_{mj}\dot{\theta}_k\dot{\boldsymbol{R}}_{i+1}^0\mathbf{z}_{mj}^{i+1} \\ &= \dot{\boldsymbol{\omega}}_i^0 + \eta_{mj}\ddot{\theta}_k\mathbf{z}_{mj}^0 + \eta_{mj}\dot{\theta}_k\overset{\times}{\boldsymbol{\omega}}{}_i^0\mathbf{z}_{mj}^0 \end{aligned} \quad (7)$$

Expressing $\dot{\boldsymbol{\omega}}_{mj}^0$ in frame $i+1$ yields:

$$\dot{\boldsymbol{\omega}}_{mj}^{i+1} = \dot{\boldsymbol{\omega}}_i^{i+1} + \eta_{mj}\ddot{\theta}_k \mathbf{z}_{mj}^{i+1} + \eta_{mj}\dot{\theta}_k \overset{\times}{\boldsymbol{\omega}}_i^{i+1}\mathbf{z}_{mj}^{i+1} \tag{8}$$

Similarly, if joint j, driven by the motor j which is attached to link i, is a rigid joint, then:

$$\dot{\boldsymbol{\omega}}_{mj}^{i+1} = \dot{\boldsymbol{\omega}}_i^{i+1} + \eta_{mj}\ddot{q}_j \mathbf{z}_{mj}^{i+1} + \eta_{mj}\dot{q}_j \overset{\times}{\boldsymbol{\omega}}_i^{i+1}\mathbf{z}_{mj}^{i+1} \tag{9}$$

Let $\mathbf{P}_{o_i}^0$ and $\mathbf{r}_{o_{i-1},i}^0$, respectively, denote the position vector of the origin of frame i and the vector pointing from $\mathbf{P}_{o_{i-1}}^0$ to $\mathbf{P}_{o_i}^0$, expressed in the base frame, then:

$$\mathbf{P}_{o_i}^0 = \mathbf{P}_{o_{i-1}}^0 + \mathbf{r}_{o_{i-1},i}^0 = \mathbf{P}_{o_{i-1}}^0 + \boldsymbol{R}_i^0 \mathbf{r}_{o_{i-1},i}^i \tag{10}$$

Notice that $\mathbf{r}_{o_{i-1},i}^i$ is a constant vector. Taking the first and second time derivatives of $\mathbf{P}_{o_i}^0$ results in:

$$\dot{\mathbf{P}}_{o_i}^0 = \dot{\mathbf{P}}_{o_{i-1}}^0 + \dot{\boldsymbol{R}}_i^0 \mathbf{r}_{o_{i-1},i}^i = \dot{\mathbf{P}}_{o_{i-1}}^0 + \overset{\times}{\boldsymbol{\omega}}_{i-1}^0 \mathbf{r}_{o_{i-1},i}^0 \tag{11}$$

$$\begin{aligned}\ddot{\mathbf{P}}_{o_i}^0 &= \ddot{\mathbf{P}}_{o_{i-1}}^0 + \overset{\times}{\dot{\boldsymbol{\omega}}}_{i-1}^0 \mathbf{r}_{o_{i-1},i}^0 + \overset{\times}{\boldsymbol{\omega}}_{i-1}^0(\dot{\boldsymbol{R}}_i^0 \mathbf{r}_{o_{i-1},i}^i) \\ &= \ddot{\mathbf{P}}_{o_{i-1}}^0 + \overset{\times}{\dot{\boldsymbol{\omega}}}_{i-1}^0 \mathbf{r}_{o_{i-1},i}^0 + \overset{\times}{\boldsymbol{\omega}}_{i-1}^0(\overset{\times}{\boldsymbol{\omega}}_{i-1}^0 \mathbf{r}_{o_{i-1},i}^0)\end{aligned} \tag{12}$$

Let $\mathbf{P}_{c_i}^0$ and $\mathbf{r}_{o_i c_i}^0$, respectively, denote the position vector of the centre of mass of the unified rigid body i and the vector pointing from $\mathbf{P}_{o_i}^0$ to $\mathbf{P}_{c_i}^0$, expressed in the base frame, then:

$$\mathbf{P}_{c_i}^0 = \mathbf{P}_{o_i}^0 + \mathbf{r}_{o_i c_i}^0 = \mathbf{P}_{o_i}^0 + \boldsymbol{R}_{i+1}^0 \mathbf{r}_{o_i c_i}^{i+1} \tag{13}$$

Notice that $\mathbf{r}_{o_i c_i}^{i+1}$ is a constant vector. Taking the first and second derivatives of $\mathbf{P}_{c_i}^0$ yields:

$$\dot{\mathbf{P}}_{c_i}^0 = \dot{\mathbf{P}}_{o_i}^0 + \overset{\times}{\boldsymbol{\omega}}_i^0 \mathbf{r}_{o_i c_i}^0 \tag{14}$$

$$\ddot{\mathbf{P}}_{c_i}^0 = \ddot{\mathbf{P}}_{o_i}^0 + \overset{\times}{\dot{\boldsymbol{\omega}}}_i^0 \mathbf{r}_{o_i c_i}^0 + \overset{\times}{\boldsymbol{\omega}}_i^0(\overset{\times}{\boldsymbol{\omega}}_i^0 \mathbf{r}_{o_i c_i}^0) \tag{15}$$

Finally, the translational accelerations $\ddot{\mathbf{P}}_{o_i}^0$ and $\ddot{\mathbf{P}}_{c_i}^0$ are, respectively, expressed in frame $i+1$ as:

$$\ddot{\mathbf{P}}_{o_i}^{i+1} = (\boldsymbol{R}_{i+1}^i)^T(\ddot{\mathbf{P}}_{o_{i-1}}^i + \overset{\times}{\dot{\boldsymbol{\omega}}}_{i-1}^i \mathbf{r}_{o_{i-1},i}^i + \overset{\times}{\boldsymbol{\omega}}_{i-1}^i(\overset{\times}{\boldsymbol{\omega}}_{i-1}^i \mathbf{r}_{o_{i-1},i}^i)) \tag{16}$$

$$\ddot{\mathbf{P}}_{c_i}^{i+1} = \ddot{\mathbf{P}}_{o_i}^{i+1} + \overset{\times}{\dot{\boldsymbol{\omega}}}_i^{i+1} \mathbf{r}_{o_i c_i}^{i+1} + \overset{\times}{\boldsymbol{\omega}}_i^{i+1}(\overset{\times}{\boldsymbol{\omega}}_i^{i+1} \mathbf{r}_{o_i c_i}^{i+1}) \tag{17}$$

B. Backward Recursion

In the backward recursion, forces and torques are propagated from the end-effector to the base link. Let $\mathbf{f}_{b,i}^0$ denote the constraint force at joint i, applied from the unified rigid body $i-1$ to the unified rigid body i. As a result, the force applied from the unified rigid body $i+1$ to the unified rigid body i is $-\mathbf{f}_{b,i+1}^0$. Newton's equation of motion for the unified rigid body i is then expressed as:

$$\mathbf{f}_{b,i}^0 - \mathbf{f}_{b,i+1}^0 + m_i \mathbf{g}^0 = m_i \ddot{\mathbf{P}}_{c_i}^0 \Leftrightarrow \mathbf{f}_{b,i}^0 = \mathbf{f}_{b,i+1}^0 + m_i(\ddot{\mathbf{P}}_{c_i}^0 - \mathbf{g}^0) \tag{18}$$

where m_i is the total mass of the unified rigid body i (including masses of link i and all motors attached to link i), $\mathbf{g}^0 = [0\ 0\ -9.81]^T\ m/s^2$ is the gravitational acceleration.

Expressing $\mathbf{f}_{b,i}^0$ in frame $i+1$ results in:

$$\mathbf{f}_{b,i}^{i+1} = \boldsymbol{R}_{i+2}^{i+1}\mathbf{f}_{b,i+1}^{i+2} + m_i(\ddot{\mathbf{P}}_{c_i}^{i+1} - \mathbf{g}^{i+1}) \tag{19}$$

Let $\mathbf{f}_e^{n+1}$ denote the external force applied to the end-effector expressed in frame $n+1$. Then the initial force for the backward recursion is:

$$\mathbf{f}_{b,n+1}^{n+1} = \mathbf{f}_e^{n+1} \tag{20}$$

Let $\bar{\boldsymbol{I}}_i^0$, $\mathbf{r}_{o_{i+1}c_i}^0$, and $\boldsymbol{\tau}_{b,i}^0$, respectively denote the inertia tensor of the unified rigid body i, the vector pointing from $\mathbf{P}_{o_{i+1}}^0$ to $\mathbf{P}_{c_i}^0$, and the torque at joint i, applied from the unified rigid body $i-1$ to the unified rigid body i. As a result, the torque applied from the unified rigid body $i+1$ to the unified rigid body i is $-\boldsymbol{\tau}_{b,i+1}^0$. Let $\sum_{\text{mr}\in\text{link i}}$ and $\sum_{\text{me}\in\text{link i}}$ denote the sum including all motors attached to link i which are, respectively, driving rigid joints and elastic joints. Note that the angular momentum of the unified rigid body i around its centre of mass is the sum of the angular momentum of link i and the angular momentum of the motors attached to link i due to their own rotation about the mass centre of the unified body. By adopting (1), Euler's equation for the unified rigid body i is:

$$\begin{aligned}\boldsymbol{\tau}_{b,i}^0 - \boldsymbol{\tau}_{b,i+1}^0 + \overset{\times}{\mathbf{f}}_{b,i}^0 \mathbf{r}_{o_i c_i}^0 - \overset{\times}{\mathbf{f}}_{b,i+1}^0 \mathbf{r}_{o_{i+1}c_i}^0 = \frac{d}{dt}\Bigg(&\bar{\boldsymbol{I}}_i^0 \boldsymbol{\omega}_i^0 \\ &+ \sum_{\text{mr}\in\text{link i}} \eta_{mj}\dot{q}_j I_{mzj}\mathbf{z}_{mj}^0 + \sum_{\text{me}\in\text{link i}} \eta_{mj}\dot{\theta}_k I_{mzj}\mathbf{z}_{mj}^0\Bigg)\end{aligned} \tag{21}$$

where it is assumed that the k^{th} elastic joint is corresponding to the j joint, which is driven by motor j.

It is worth noting that, when expressing $\bar{\boldsymbol{I}}_i^0$ in frame $i+1$, the inertia tensor of the unified rigid body i becomes constant. Furthermore, $\bar{\boldsymbol{I}}_i^0 = \boldsymbol{R}_{i+1}^0 \bar{\boldsymbol{I}}_i^{i+1}(\boldsymbol{R}_{i+1}^0)^T$. Then, the first term on the right-hand side of (21) is written as:

$$\begin{aligned}\frac{d}{dt}\left(\bar{\boldsymbol{I}}_i^0 \boldsymbol{\omega}_i^0\right) &= \frac{d}{dt}\left(\boldsymbol{R}_{i+1}^0 \bar{\boldsymbol{I}}_i^{i+1}(\boldsymbol{R}_{i+1}^0)^T \boldsymbol{\omega}_i^0\right) \\ &= \bar{\boldsymbol{I}}_i^0 \dot{\boldsymbol{\omega}}_i^0 + \dot{\boldsymbol{R}}_{i+1}^0 \bar{\boldsymbol{I}}_i^{i+1}(\boldsymbol{R}_{i+1}^0)^T \boldsymbol{\omega}_i^0 + \boldsymbol{R}_{i+1}^0 \bar{\boldsymbol{I}}_i^{i+1}(\dot{\boldsymbol{R}}_{i+1}^0)^T \boldsymbol{\omega}_i^0 \\ &= \bar{\boldsymbol{I}}_i^0 \dot{\boldsymbol{\omega}}_i^0 + \overset{\times}{\boldsymbol{\omega}}_i^0(\bar{\boldsymbol{I}}_i^0 \boldsymbol{\omega}_i^0) + \bar{\boldsymbol{I}}_i^0 \overset{\times}{\boldsymbol{\omega}}_i^0 \boldsymbol{\omega}_i^0 \\ &= \bar{\boldsymbol{I}}_i^0 \dot{\boldsymbol{\omega}}_i^0 + \overset{\times}{\boldsymbol{\omega}}_i^0(\bar{\boldsymbol{I}}_i^0 \boldsymbol{\omega}_i^0)\end{aligned} \tag{22}$$

The last two terms on the right-hand side of (21) can also be simplified as:

$$\begin{aligned}&\frac{d}{dt}\left(\sum_{\text{mr}\in\text{link i}} \eta_{mj}\dot{q}_j I_{mzj}\mathbf{z}_{mj}^0 + \sum_{\text{me}\in\text{link i}} \eta_{mj}\dot{\theta}_k I_{mzj}\mathbf{z}_{mj}^0\right) \\ &= \frac{d}{dt}\Bigg(\sum_{\text{mr}\in\text{link i}} \eta_{mj}\dot{q}_j I_{mzj}\boldsymbol{R}_{i+1}^0 \mathbf{z}_{mj}^{i+1} \\ &\qquad + \sum_{\text{me}\in\text{link i}} \eta_{mj}\dot{\theta}_k I_{mzj}\boldsymbol{R}_{i+1}^0 \mathbf{z}_{mj}^{i+1}\Bigg) \\ &= \sum_{\text{mr}\in\text{link i}} \left(\eta_{mj}\ddot{q}_j I_{mzj}\mathbf{z}_{mj}^0 + \eta_{mj}\dot{q}_j I_{mzj}\overset{\times}{\boldsymbol{\omega}}_i^0 \mathbf{z}_{mj}^0\right) \\ &\qquad + \sum_{\text{me}\in\text{link i}} \left(\eta_{mj}\ddot{\theta}_k I_{mzj}\mathbf{z}_{mj}^0 + \eta_{mj}\dot{\theta}_k I_{mzj}\overset{\times}{\boldsymbol{\omega}}_i^0 \mathbf{z}_{mj}^0\right)\end{aligned} \tag{23}$$

Substituting (22) and (23) into Euler's equation (21) yields:

$$\begin{aligned}
\boldsymbol{\tau}_{b,i}^0 &= \boldsymbol{\tau}_{b,i+1}^0 - \overset{\times}{\mathbf{f}}{}_{b,i}^0 \mathbf{r}_{o_i c_i}^0 + \overset{\times}{\mathbf{f}}{}_{b,i+1}^0 \mathbf{r}_{o_{i+1} c_i}^0 \\
&+ \bar{\boldsymbol{I}}_i^0 \dot{\boldsymbol{\omega}}_i^0 + \overset{\times}{\boldsymbol{\omega}}{}_i^0 (\bar{\boldsymbol{I}}_i^0 \boldsymbol{\omega}_i^0) \\
&+ \sum_{\text{mr}\in\text{link i}} \Big(\eta_{mj} \ddot{q}_j I_{mzj} \mathbf{z}_{mj}^0 + \eta_{mj} \dot{q}_j I_{mzj} \overset{\times}{\boldsymbol{\omega}}{}_i^0 \mathbf{z}_{mj}^0 \Big) \\
&+ \sum_{\text{me}\in\text{link i}} \Big(\eta_{mj} \ddot{\theta}_k I_{mzj} \mathbf{z}_{mj}^0 + \eta_{mj} \dot{\theta}_k I_{mzj} \overset{\times}{\boldsymbol{\omega}}{}_i^0 \mathbf{z}_{mj}^0 \Big)
\end{aligned} \tag{24}$$

Expressing $\boldsymbol{\tau}_{b,i}^0$ in frame $i+1$ yields:

$$\begin{aligned}
\boldsymbol{\tau}_{b,i}^{i+1} &= \boldsymbol{R}_{i+2}^{i+1} \boldsymbol{\tau}_{b,i+1}^{i+2} - \overset{\times}{\mathbf{f}}{}_{b,i}^{i+1} \mathbf{r}_{o_i c_i}^{i+1} + [\boldsymbol{R}_{i+2}^{i+1} \mathbf{f}_{b,i+1}^{i+2}]_\times \mathbf{r}_{o_{i+1} c_i}^{i+1} \\
&+ \bar{\boldsymbol{I}}_i^{i+1} \dot{\boldsymbol{\omega}}_i^{i+1} + [\boldsymbol{\omega}_i^{i+1}]_\times (\bar{\boldsymbol{I}}_i^{i+1} \boldsymbol{\omega}_i^{i+1}) \\
&+ \sum_{\text{mr}\in\text{link i}} \big(\eta_{mj} \ddot{q}_j I_{mzj} \mathbf{z}_{mj}^{i+1} + \eta_{mj} \dot{q}_j I_{mzj} [\boldsymbol{\omega}_i^{i+1}]_\times \mathbf{z}_{mj}^{i+1} \big) \\
&+ \sum_{\text{me}\in\text{link i}} \Big(\eta_{mj} \ddot{\theta}_k I_{mzj} \mathbf{z}_{mj}^{i+1} + \eta_{mj} \dot{\theta}_k I_{mzj} [\boldsymbol{\omega}_i^{i+1}]_\times \mathbf{z}_{mj}^{i+1} \Big)
\end{aligned} \tag{25}$$

Let $\boldsymbol{\tau}_e^{n+1}$ denote the external torque applied to the end-effector expressed in frame $n+1$. Then the initial torque for the backward recursion is:

$$\boldsymbol{\tau}_{b,n+1}^{n+1} = \boldsymbol{\tau}_e^{n+1} \tag{26}$$

Let $\boldsymbol{\tau} = [\tau_1 \ ... \ \tau_N]^T$ denote the vector of generalised forces, which are in fact the motor torques applied to the corresponding joint. By summing up the moment $\boldsymbol{\tau}_{b,i}^0$ projected along the rotational axis $\mathbf{z}_i^0$ of the joint and the inertia torque of motor i, the generalised force τ_i at joint i is obtained. Nevertheless, $\boldsymbol{\tau}_{b,i}^0$ depends on the type of the joint. Particularly, if joint i is a rigid joint, then:

$$\begin{aligned}
\tau_i &= (\boldsymbol{\tau}_{b,i}^0)^T \mathbf{z}_i^0 + \eta_{mi} I_{mzi} (\dot{\boldsymbol{\omega}}_{mi}^0)^T \mathbf{z}_{mi}^0 \\
&= (\boldsymbol{\tau}_{b,i}^{i+1})^T (\boldsymbol{R}_{i+1}^i)^T \mathbf{z}_0 + \eta_{mi} I_{mzi} (\dot{\boldsymbol{\omega}}_{mi}^{l_{mi}})^T \mathbf{z}_{mi}^{l_{mi}}
\end{aligned} \tag{27}$$

On the other hand, the motor side and the load side of an elastic joint are not directly connected but through an elastic torque. Let K_k denote the stiffness of the k^{th} elastic joint. Then, if joint i is the k^{th} elastic joint, the load side dynamics is described as follows:

$$(\boldsymbol{\tau}_{b,i}^0)^T \mathbf{z}_i^0 = K_k(\theta_k - q_i) \Leftrightarrow (\boldsymbol{\tau}_{b,i}^{i+1})^T (\boldsymbol{R}_{i+1}^i)^T \mathbf{z}_0 = K_k(\theta_k - q_i) \tag{28}$$

and the motor dynamics is described as follows:

$$\begin{aligned}
\tau_i &= K_k(\theta_k - q_i) + \eta_{mi} I_{mzi} \dot{\boldsymbol{\omega}}_{mi}^{0\,T} \mathbf{z}_{mi}^0 \\
&= K_k(\theta_k - q_i) + \eta_{mi} I_{mzi} \dot{\boldsymbol{\omega}}_{mi}^{l_{mi}\,T} \mathbf{z}_{mi}^{l_{mi}}
\end{aligned} \tag{29}$$

The recursive Newton-Euler algorithm is summarised in the Algorithm 1, where $\mathbf{f}_b = [\mathbf{f}_{b,1}^2 \ ... \ \mathbf{f}_{b,N}^{N+1}]$ and $\boldsymbol{\tau}_b = [\boldsymbol{\tau}_{b,1}^2 \ ... \ \boldsymbol{\tau}_{b,N}^{N+1}]$ are the data sets collecting the interaction forces and torques.

Algorithm 1 Recursive NE Algorithm Inverse Dynamics

function RENEAL
  Input: $[\mathbf{q}^T \ \boldsymbol{\theta}^T]^T$, $[\dot{\mathbf{q}}^T \ \dot{\boldsymbol{\theta}}^T]^T$, $[\ddot{\mathbf{q}}^T \ \ddot{\boldsymbol{\theta}}^T]^T$, $\mathbf{f}_e^{n+1}$, $\boldsymbol{\tau}_e^{n+1}$, $\mathbf{g}^0$, $\boldsymbol{\omega}_0^0$, $\dot{\boldsymbol{\omega}}_0^0$, $\ddot{\mathbf{P}}_{o_0}^0$
  Output: $\boldsymbol{\tau}$, $\mathbf{f}_b$, $\boldsymbol{\tau}_b$

  FORWARD KINEMATICS
  for $i \leftarrow 1$ to N **do** (4), (5), (16), (17)
  end for
  for $i \leftarrow 1$ to N **do**
    if joint i is rigid **then** (8)
    else if joint i is the k^{th} elastic joint **then** (9)
    end if
  end for

  BACKWARD RECURSION
  for $i \leftarrow N$ to 1 **do** (19), (25)
  end for

  GENERALISED TORQUE
  for $i \leftarrow 1$ to N **do**
    if joint i is rigid **then** (27)
    else if joint i is the k^{th} elastic joint **then** (29)
    end if
  end for

  gather components of $\boldsymbol{\tau}$, $\mathbf{f}_b$, and $\boldsymbol{\tau}_b$
  return $\boldsymbol{\tau}$, $\mathbf{f}_b$, $\boldsymbol{\tau}_b$
end function

III. ALGORITHM FOR COMPUTING FORWARD DYNAMICS

For simulating the dynamics of robots with mixed rigid-elastic joints, forward dynamics is computed, where the joint accelerations are determined given the joint positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, and motor torques $\boldsymbol{\tau}_m$. First of all, the compact form of dynamic equations of robots with mixed rigid-elastic joints is written as follows [7]:

$$\begin{bmatrix} \boldsymbol{M}(\mathbf{q}) & \boldsymbol{S}(\mathbf{q}) \\ \mathbf{S}(\mathbf{q})^T & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{C}_1(\mathbf{q}, \dot{\mathbf{q}})\dot{\boldsymbol{\theta}} \\ \boldsymbol{C}_2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{G}(\mathbf{q}) \\ \mathbf{U}_\theta \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_q + \boldsymbol{\tau}_{ex} + \boldsymbol{\tau}_{fr,l} \\ \boldsymbol{\tau}_{me} + \boldsymbol{\tau}_{fr,m} \end{bmatrix} \tag{30}$$

where $\boldsymbol{M}(\mathbf{q})$ is the $N \times N$ matrix , $\boldsymbol{S}(\mathbf{q})$ is the $N \times N_e$ coupling matrix, $\boldsymbol{B}$ is the $N_e \times N_e$ inertia matrix, $\boldsymbol{C}(\mathbf{q}, \dot{\mathbf{q}})$, $\boldsymbol{C}_1(\mathbf{q}, \dot{\mathbf{q}})$, $\boldsymbol{C}_2(\mathbf{q}, \dot{\mathbf{q}})$ are coriolis and centrifugal matrices, $\mathbf{G}(\mathbf{q})$ is the gravitational vector, $\mathbf{U}_\theta$ is the vector of elastic forces, $\boldsymbol{\tau}_q$ is the vector including forces at the joints, $\boldsymbol{\tau}_{me}$ is the vector including motor torques of the elastic joints, $\boldsymbol{\tau}_{ex}$ is the interaction torques, and $\boldsymbol{\tau}_{fr,l}$, $\boldsymbol{\tau}_{fr,m}$ are friction torques, while:

$$\boldsymbol{\tau}_q = [\tau_{q_1} \ ... \ \tau_{q_N}]^T$$

$$\tau_{q_i} = \begin{cases} K_k(\theta_k - q_i) & \text{if joint } i \text{ is the } k^{th} \text{ elastic joint} \\ \tau_{mr_i}(t) & \text{if joint } i \text{ is a rigid joint} \end{cases}$$

where $\tau_{mri}(t)$ is the motor torque of the rigid joint i.

Then, we rewrite (30) as:

$$\boldsymbol{A}(\mathbf{q})\begin{bmatrix}\ddot{\mathbf{q}}\\ \ddot{\theta}\end{bmatrix}+\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})=\begin{bmatrix}\boldsymbol{\tau}_q+\boldsymbol{\tau}_{fr,l}\\ \boldsymbol{\tau}_{me}+\boldsymbol{\tau}_{fr,m}\end{bmatrix} \tag{31}$$

where

$$\boldsymbol{A}(\mathbf{q})=\begin{bmatrix}\boldsymbol{M}(\mathbf{q}) & \boldsymbol{S}(\mathbf{q})\\ \boldsymbol{S}(\mathbf{q})^T & \boldsymbol{B}\end{bmatrix}$$

$$\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})=\begin{bmatrix}\boldsymbol{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\boldsymbol{C}_1(\mathbf{q},\dot{\mathbf{q}})\dot{\theta}\\ \boldsymbol{C}_2(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\end{bmatrix}+\begin{bmatrix}\mathbf{G}(\mathbf{q})-\boldsymbol{\tau}_{ex}\\ \mathbf{U}_\theta\end{bmatrix}$$

In the robotic community, mainly, the Lagrangian-based method is used to determine the equations of motion. However, as the system becomes complex, the calculation of known quantities such as the mass matrix could become prohibited computationally. In the following, the recursive Newton-Euler algorithm is adopted to extract the mass matrix. Firstly, the recursive Newton-Euler algorithm is run with $[\ddot{\mathbf{q}}^T\ \ddot{\theta}^T]^T=\mathbf{0}^T$. Then, the $(N+N_e)\times 1$ vector $\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})=[\zeta_1\ \ldots\ \zeta_N\ \zeta_{N+1}\ \ldots\ \zeta_{N+N_e}]^T$ can be determined. For the first N elements of vector $\zeta(q,\dot{q},\theta,\dot{\theta})$:

$$\zeta_i=\begin{cases}(\boldsymbol{\tau}_{b,i}^{i+1})^T(\boldsymbol{R}_{i+1}^i)^T\mathbf{z}_0 & \text{if joint } i \text{ is an elastic joint}\\ \tau_i & \text{if joint } i \text{ is a rigid joint}\end{cases} \tag{32}$$

and for the last N_e elements of vector $\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})$, if joint i is the k^{th} elastic joint, then:

$$\zeta_{N+k}=\tau_i \tag{33}$$

Secondly, the $(N+N_e)\times(N+N_e)$ matrix $\boldsymbol{A}(\mathbf{q})$ will be constructed numerically. To determine column j, the recursive Newton-Euler algorithm is performed, with $[\dot{\mathbf{q}}^T\dot{\theta}^T]^T=\mathbf{0}^T$, $\mathbf{f}_e^{n+1}=\mathbf{0}^T$, $\boldsymbol{\tau}_e^{n+1}=\mathbf{0}^T$, $\mathbf{g}^0=\mathbf{0}^T$, $[\ddot{\mathbf{q}}^T\ddot{\theta}^T]^T=\mathbf{I}_j^T$, and $\mathbf{I}_j$ is the j^{th} column of an identity matrix. Then, for the first N elements of column A_j:

$$A_{i,j}=\begin{cases}(\boldsymbol{\tau}_{b,i}^{i+1})^T(\boldsymbol{R}_{i+1}^i)^T\mathbf{z}_0 & \text{if joint } i \text{ is an elastic joint}\\ \tau_i & \text{if joint } i \text{ is a rigid joint}\end{cases} \tag{34}$$

and for the last N_e elements of column A_j, if joint i is the k^{th} elastic joint, then:

$$A_{N+k,j}=\tau_i-K_k(\theta_k-q_i) \tag{35}$$

Lastly, the accelerations are calculated as follows:

$$\begin{bmatrix}\ddot{\mathbf{q}}\\ \ddot{\theta}\end{bmatrix}=\boldsymbol{A}(\mathbf{q})^{-1}\left(\begin{bmatrix}\boldsymbol{\tau}_q+\boldsymbol{\tau}_{fr,l}\\ \boldsymbol{\tau}_{me}+\boldsymbol{\tau}_{fr,m}\end{bmatrix}-\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})\right) \tag{36}$$

The process is implemented as shown in Algorithm 2.

Algorithm 2 Forward dynamics calculation

function DIRDYN
  Input: $[\mathbf{q}^T\ \theta^T]^T$, $[\dot{\mathbf{q}}^T\ \dot{\theta}^T]^T$, $\boldsymbol{\tau}_m$, $\mathbf{f}_e^{n+1}$, $\boldsymbol{\tau}_e^{n+1}$
  Output: $[\ddot{\mathbf{q}}^T\ \ddot{\theta}^T]^T$

  CALCULATE $\zeta(\mathbf{q},\dot{\mathbf{q}},\theta,\dot{\theta})$
  $[\boldsymbol{\tau},\ \mathbf{f}_b,\ \boldsymbol{\tau}_b]$ = ReNEAl($[\mathbf{q}^T\ \theta^T]^T$, $[\dot{\mathbf{q}}^T\ \dot{\theta}^T]^T$, $\mathbf{0}^T$, $\mathbf{f}_e^{n+1}$, $\boldsymbol{\tau}_e^{n+1}$, $\mathbf{g}^0$, ω_0^0, $\dot{\omega}_0^0$, $\ddot{\mathbf{P}}_{o_0}^0$)
  for $i\leftarrow 1$ to N **do**
    if joint i is rigid **then**
      $\zeta_i=\tau_i$
    else if joint i is the k^{th} elastic joint **then**
      $\zeta_i=(\boldsymbol{\tau}_{b,i}^{i+1})^T(\boldsymbol{R}_{i+1}^i)^T\mathbf{z}_0$
      $\zeta_{N+k}=\tau_i$
    end if
  end for

  CALCULATE $\boldsymbol{A}(\mathbf{q})$
  for $j\leftarrow 1$ to $N+N_e$ **do**
    $[\boldsymbol{\tau},\ \mathbf{f}_b,\ \boldsymbol{\tau}_b]$ = ReNEAl($[\mathbf{q}^T\ \theta^T]^T$, $\mathbf{0}^T$, $\mathbf{I}_j$, $\mathbf{0}^T$, $\mathbf{0}^T$, $\mathbf{0}^T$, ω_0^0, $\dot{\omega}_0^0$, $\ddot{\mathbf{P}}_{o_0}^0$)
    for $i\leftarrow 1$ to N **do**
      if joint i is rigid **then**
        $A_{i,j}=\tau_i$
      else if joint i is the k^{th} elastic joint **then**
        $A_{i,j}=(\boldsymbol{\tau}_{b,i}^{i+1})^T(\boldsymbol{R}_{i+1}^i)^T\mathbf{z}_0$
        $A_{N+k,j}=\tau_i-K_k(\theta_k-q_i)$
      end if
    end for
  end for

  calculate $[\ddot{\mathbf{q}}^T\ \ddot{\theta}^T]^T$
  Calculate $\boldsymbol{\tau}_q$
  for $i\leftarrow 1$ to N **do**
    if joint i is the k^{th} elastic joint **then**
      $\tau_{q_i}=K_k\eta_{li}(\theta_k-\eta_{li}q_i)$
    else if joint i is a rigid joint **then**
      $\tau_{q_i}=\tau_{mri}(t)$
    end if
  end for
  Calculate $\ddot{\mathbf{q}}$ and $\ddot{\theta}$ using (36)
  return $[\ddot{\mathbf{q}}^T\ \ddot{\theta}^T]^T$
end function

IV. INVERSE DYNAMICS CONTROLLER

This section introduces an inverse dynamics control algorithm aimed at controlling robots equipped with a combination of rigid and elastic joints. Let $\mathbf{q}_d$ and $\mathbf{e}_{qd}=\mathbf{q}_d-\mathbf{q}$, respectively, denote the desired load position vector and the error between the $\mathbf{q}_d$ and the actual load position. Let $\boldsymbol{K}_{Pq}$ and $\boldsymbol{K}_{Dq}$ denote positive diagonal matrices equivalent to proportional and derivative coefficients in a PD controller, respectively. The reference velocity is defined as:

$$\dot{\mathbf{q}}_r=\dot{\mathbf{q}}_d+\boldsymbol{K}_{Dq}^{-1}\boldsymbol{K}_{Pq}\mathbf{e}_{qd}=\dot{\mathbf{q}}_d+\boldsymbol{\Lambda}_q\mathbf{e}_{qd} \tag{37}$$
$$\boldsymbol{\Lambda}_q=\boldsymbol{K}_{Dq}^{-1}\boldsymbol{K}_{Pq}$$

and the error between the reference velocity and the actual velocity is defined as:

$$\boldsymbol{\sigma}_q=\dot{\mathbf{q}}_r-\dot{\mathbf{q}}=\dot{\mathbf{e}}_{qd}+\boldsymbol{\Lambda}_q\mathbf{e}_{qd} \tag{38}$$

Then, the control law for the load side is proposed:

$$\boldsymbol{\tau}_{qc} = \boldsymbol{M}(\mathbf{q})\ddot{\mathbf{q}}_r + \boldsymbol{S}(\mathbf{q})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \boldsymbol{C}_1(\mathbf{q},\dot{\mathbf{q}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{K}_{Dq}\boldsymbol{\sigma}_q \tag{39}$$

Subtracting the dynamic equation of the load side in (30) from the control law (39) gives:

$$\boldsymbol{M}(\mathbf{q})\dot{\boldsymbol{\sigma}}_q + \boldsymbol{C}(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\sigma}_q + \boldsymbol{K}_{Dq}\boldsymbol{\sigma}_q = \mathbf{e}_{\tau_q} \tag{40}$$

where

$$\mathbf{e}_{\tau_q} = \boldsymbol{\tau}_{qc} - \boldsymbol{\tau}_q = [e_{\tau_q 1} \ ... e_{\tau_q N}]^T$$

$$e_{\tau_q i} = \begin{cases} K_k(\theta_{dk} - \theta_k) & \text{if joint } i \text{ is the } k^{th} \text{ elastic joint} \\ 0 & \text{if joint } i \text{ is a rigid joint} \end{cases}$$

and θ_{dk} is the desired motor angular position.

We now introduce a Lyapunov function for the load:

$$V_1 = \frac{1}{2}\boldsymbol{\sigma}_q^T \boldsymbol{M}(\mathbf{q})\boldsymbol{\sigma}_q + \frac{1}{2}\mathbf{e}_{qd}^T 2\boldsymbol{\Lambda}_q^T \boldsymbol{K}_{Dq}\mathbf{e}_{qd} \tag{41}$$

Taking time-derivative of V_1 and applying (40), with the skew-symmetry property of the matrix $\dot{\boldsymbol{M}}(\mathbf{q}) - 2\boldsymbol{C}(\mathbf{q},\dot{\mathbf{q}})$, results in:

$$\begin{aligned}\dot{V}_1 &= \boldsymbol{\sigma}_q^T \boldsymbol{M}(\mathbf{q})\dot{\boldsymbol{\sigma}}_q + \mathbf{e}_{qd}^T 2\boldsymbol{\Lambda}_q^T \boldsymbol{K}_{Dq}\dot{\mathbf{e}}_{qd} \\ &= -\dot{\mathbf{e}}_{qd}^T \boldsymbol{K}_{Dq}\dot{\mathbf{e}}_{qd} - \mathbf{e}_{qd}^T \boldsymbol{\Lambda}_q^T \boldsymbol{K}_{Dq}\boldsymbol{\Lambda}_q \mathbf{e}_{qd} + \boldsymbol{\sigma}_q^T \mathbf{e}_{\tau_q}\end{aligned} \tag{42}$$

From equation (39), the desired motor angular positions for the elastic joints are determined as follows:

$$\tau_{qc_i} = K_k(\theta_{dk} - q_i) \Rightarrow \theta_{dk} = \frac{1}{K_k}\tau_{qc_i} + q_i \tag{43}$$

Let $\boldsymbol{\theta}_d$ denote the vector representing the desired angular positions of the elastic joint motors. In addition, let $\boldsymbol{\sigma}_{qe}$ denote the vector containing only elements related to the elastic joints from $\boldsymbol{\sigma}_q$. Next, the errors for the motors are defined analogously to those for the joints:

$$\mathbf{e}_{\theta d} = \boldsymbol{\theta}_d - \boldsymbol{\theta} \tag{44}$$

$$\dot{\boldsymbol{\theta}}_r = \dot{\boldsymbol{\theta}}_d + \boldsymbol{K}_{D\theta}^{-1}\boldsymbol{K}_{P\theta}\mathbf{e}_{\theta d} + \boldsymbol{\sigma}_{qe} = \dot{\boldsymbol{\theta}}_d + \boldsymbol{\Lambda}_{\theta}\mathbf{e}_{\theta d} + \boldsymbol{\sigma}_{qe} \tag{45}$$

$$\boldsymbol{\Lambda}_\theta = \boldsymbol{K}_{D\theta}^{-1}\boldsymbol{K}_{P\theta} \tag{46}$$

$$\boldsymbol{\sigma}_\theta = \dot{\boldsymbol{\theta}}_r - \dot{\boldsymbol{\theta}} = \dot{\mathbf{e}}_{\theta d} + \boldsymbol{\Lambda}_\theta \mathbf{e}_{\theta d} + \boldsymbol{\sigma}_{qe} \tag{47}$$

It is worth noting that if joint i is rigid, its component in the vector $\mathbf{e}_{\tau_q}$ is 0, hence:

$$\boldsymbol{\sigma}_q^T \mathbf{e}_{\tau_q} = \boldsymbol{\sigma}_{qe}^T \boldsymbol{K}\mathbf{e}_{\theta d} \tag{48}$$

The control law for the motor side is proposed:

$$\boldsymbol{\tau}_{me} = \boldsymbol{S}(\mathbf{q})^T\ddot{\mathbf{q}} + \boldsymbol{B}\ddot{\boldsymbol{\theta}}_r + \boldsymbol{C}_2(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{K}(\boldsymbol{\theta}_d - \mathbf{q}_e) + \boldsymbol{K}_{D\theta}\boldsymbol{\sigma}_\theta \tag{49}$$

Substituting (49) into (30) gives:

$$\boldsymbol{B}\dot{\boldsymbol{\sigma}}_\theta + \boldsymbol{K}_{D\theta}\boldsymbol{\sigma}_\theta + \boldsymbol{K}\mathbf{e}_{\theta d} = 0 \tag{50}$$

Introduce a Lyapunov function including the errors of the motor side:

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{\sigma}_\theta^T \boldsymbol{B}\boldsymbol{\sigma}_\theta + \frac{1}{2}\mathbf{e}_{\theta d}^T \boldsymbol{K}\mathbf{e}_{\theta d} \tag{51}$$

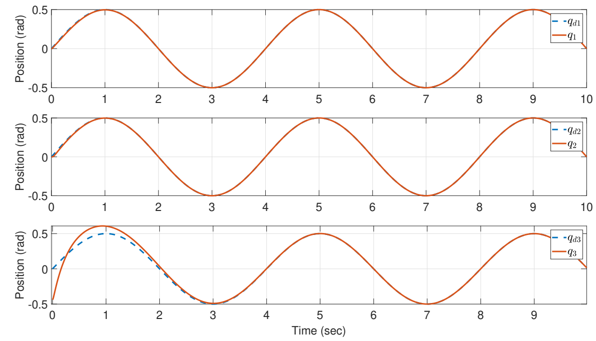


Fig. 3: Load angular positions when there is no payload.

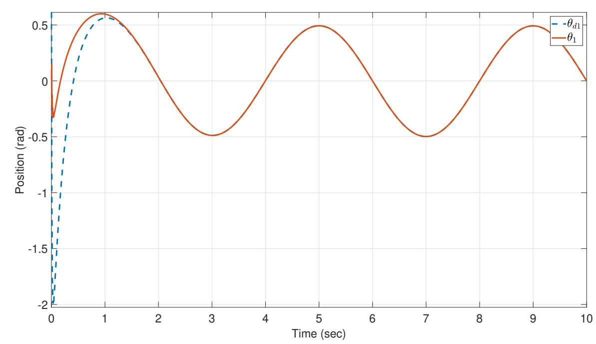


Fig. 4: Motor angular position when there is no payload.

Taking time-derivative of V_2 and applying (50) yields:

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \boldsymbol{\sigma}_\theta^T \boldsymbol{B}\dot{\boldsymbol{\sigma}}_\theta + \mathbf{e}_{\theta d}^T \boldsymbol{K}\dot{\mathbf{e}}_{\theta d} \\ &= -\dot{\mathbf{e}}_{qd}^T \boldsymbol{K}_{Dq}\dot{\mathbf{e}}_{qd} - \mathbf{e}_{qd}^T \boldsymbol{\Lambda}_q^T \boldsymbol{K}_{Dq}\boldsymbol{\Lambda}_q \mathbf{e}_{qd} \\ &\quad - \boldsymbol{\sigma}_\theta^T \boldsymbol{K}_{D\theta}\boldsymbol{\sigma}_\theta - \mathbf{e}_{\theta d}^T \boldsymbol{K}\boldsymbol{\Lambda}_\theta \mathbf{e}_{\theta d} < 0\end{aligned} \tag{52}$$

Therefore, the system is stable.

V. Simulation

In this section, the forward dynamics algorithm 2 is used to simulate a robot with mixed rigid-elastic joints. The simulated robot has 3 DOFs, in which the first 2 joints are rigid and the last one is elastic. The stiffness of the spring of joint 3 is 100 (N/m). Motor 1 is attached to the base link 0, and motors 2 and 3 are attached to link 1. The dynamic parameters of the robot are mentioned in Table I. The desired positions of the joints are sinusoidal functions of the form $0.5\sin(0.5\pi t)$. The inverse dynamics controller is adopted for tracking the desired position. In addition, the robot is simulated with and without a payload attached to the end effector within 10 seconds.

TABLE I: Dynamic Parameters

	link 1	link 2
mass(kg)	2.5	1.5
inertia (kg.m^2)	$\begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.35 & 0.15 & 0.1 \\ 0.15 & 0.4 & 0.2 \\ 0.1 & 0.2 & 0.3 \end{bmatrix}$
	link 3	motor 1, 2, 3
mass(kg)	1.0	1.0
inertia (kg.m^2)	$\begin{bmatrix} 0.3 & 0.15 & 0.13 \\ 0.15 & 0.25 & 0.2 \\ 0.13 & 0.2 & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.006 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.004 \end{bmatrix}$

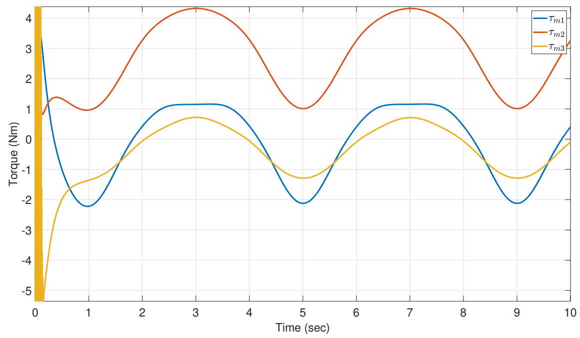


Fig. 5: Motor torque when there is no payload.

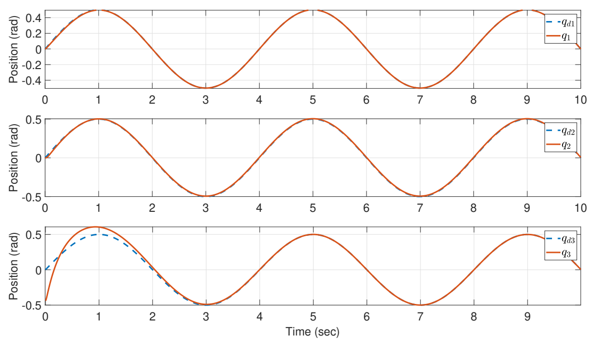


Fig. 6: Load angular positions when there is a payload.

Figures 3 and 4 respectively show the load and motor angular positions of the robot when there is no external payload. The initial values of the load positions are $[0\ 0\ -0.4363]^T$ radians. The controller can manage to track the desired positions, and the root mean square errors of three joints, respectively, are 0.0026819 radians, 0.003619 radians, and 0.052678 radians. Figure 5 illustrates the motor torques. The large oscillation of the motor torque of joint 3 at the beginning corresponds with the time when the motor tries to track the desired motor position of the elastic joint.

To simulate the effect of payload on the robot with mixed rigid-elastic joints, the external force in Algorithm 2 is set to $\mathbf{f}_e^0 = [0\ 0\ \ -3]^T$ Newton. Figures 6 and 7, respectively, show the load and motor angular positions of the robot in this case. The initial values of the load positions are also $[0\ 0\ \ -0.4363]^T$ radians. The controller can also track the desired positions, and the root mean square errors of three joints, respectively, are 0.0026903 radians, 0.0061066 radians,

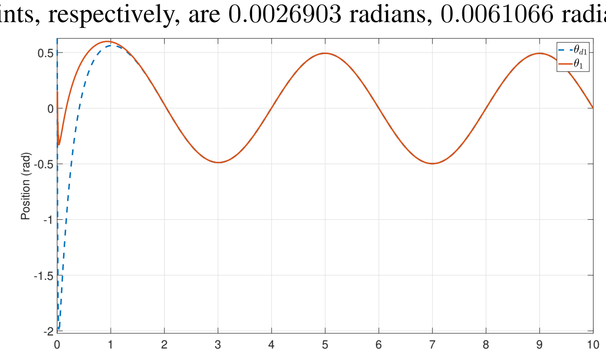


Fig. 7: Motor angular position when there is a payload.

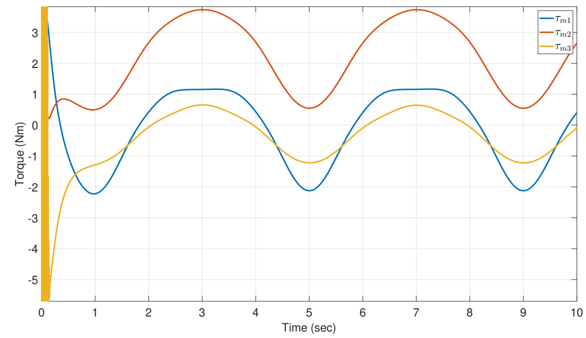


Fig. 8: Motor torque when there is a payload.

and 0.052422 radians. Figure 8 illustrates the motor torques when there is an external payload. It is clear that the motor torques are lower, for example, when the robot moves to the lowest position at 3 seconds because payload's weight.

VI. Conclusion

In this paper, a recursive Newton-Euler algorithm to compute the inverse dynamics of robots with mixed rigid-elastic joints is proposed. In addition, a forward dynamics algorithm to compute the accelerations given positions, velocities and motor torques is presented. Furthermore, an inverse dynamics control algorithm is introduced to control robots with mixed joint types. Finally, the mathematical modelling method is validated through simulations with and without a payload attached to the end effector of the robot. It is shown that the mathematical model works as expected. In subsequent research endeavors, a focus will be directed towards the development of an adaptive controller tailored to effectively control the robot's interactions with its surrounding environment.

References

[1] Gabriele Buondonno and Alessandro De Luca. A recursive Newton-Euler algorithm for robots with elastic joints and its application to control. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5526–5532. IEEE, 2015.

[2] Saeed Zaare and Mohammad Reza Soltanpour. Adaptive fuzzy global coupled nonsingular fast terminal sliding mode control of n-rigid-link elastic-joint robot manipulators in presence of uncertainties. *Mechanical Systems and Signal Processing*, 163:108165, 2022.

[3] Tuan Minh Hua, Filippo Sanfilippo, and Erlend Helgerud. A robust two-feedback loops position control algorithm for compliant low-cost series elastic actuators. In *Proc. of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2384–2390, 2019.

[4] Hua Minh Tuan, Filippo Sanfilippo, and Nguyen Vinh Hao. Modelling and control of a 2-dof robot arm with elastic joints for safe human-robot interaction. *Frontiers in Robotics and AI*, 8:679304, 2021.

[5] Alessandro De Luca. Decoupling and feedback linearization of robots with mixed rigid/elastic joints. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 8(11):965–977, 1998.

[6] Wen-Hong Zhu and Joris De Schutter. Adaptive control of mixed rigid/flexible joint robot manipulators based on virtual decomposition. *IEEE Transactions on Robotics and Automation*, 15(2):310–317, 1999.

[7] A De Luca and R Farina. Dynamic properties and nonlinear control of robots with mixed rigid/elastic joints. In *Proceedings World Automation Congress, 2004.*, volume 15, pages 97–104. IEEE, 2004.

[8] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009.