Dynamic Modeling and Modal Analysis of Multiple Degrees of Freedom Robot Manipulators with Flexible Joints

by

Thanh Trung DO

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, DECEMBER 13, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Zhaoheng Liu, Thesis supervisor
Department of Mechanical Engineering, École de technologie supérieure

Mr. Viet-Hung Vu, Thesis Co-Supervisor
Department of Mechanical Engineering, École de technologie supérieure

Mr. Ilian Bonev, Chair, Board of Examiners
Department of Systems Engineering, École de technologie supérieure

Mr. Jean-Pierre Kenné, Member of the Jury
Department of Mechanical Engineering, École de technologie supérieure

Mr. Marc Arsenault, External Examiner
Bharti School of Engineering, Laurentian University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON MARCH 3, 2022

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Professor Zhaoheng Liu and my co-supervisor, Doctor Viet-Hung Vu, for their guidance, encouragement and support during my years of study at ÉTS. Without their encouragement, this thesis could not have been completed.

I would like to thank my PhD committee members: Professor Ilian Bonev, Professor Jean-Pierre Kenné, and Professor Marc Arsenault for accepting to review this work as well as giving me many valuable comments.

I also wish to take this opportunity to thank the professors in the Department of Mechanical Engineering at ÉTS for awarding me the scholarship at the final stage of my study.

Finally, I would like to express my warmest thanks to my parents, my parents-in-law, my sisters, and my sisters/brothers-in-law for their support, encouragement and assistance. Especially, I would like to thank my wife for her love, care, and support. A special thanks goes to my children for being with me on this journey.

Modélisation Dynamique et Analyse Modale de Robots Manipulateurs à Plusieurs Degrés de Liberté avec Articulations Flexibles

Thanh Trung DO

RÉSUMÉ

En raison de leur flexibilité dans la production et du fait de leur interaction sécuritaire avec l'humain, les robots à maillons rigides et à articulations flexibles (élastiques) ont reçu récemment beaucoup d'attention de la part des chercheurs et des ingénieurs au milieu d'application. Auparavant, l'élasticité des articulations était souvent négligée à cause de la complexité des modèles dynamiques. En outre, les limites de la capacité à mesurer et à calculer la dynamique ont fait en sorte que le modèle de robot avec des articulations rigides soit souvent utilisé à des fins de simulation et de contrôle. Cette approche peut cependant entraîner l'instabilité du robot pendant le travail.

Le premier objectif de cette recherche est de développer des modèles mathématiques pertinents pour les robots à articulations flexibles afin de simuler et de prédire leur comportement dynamique. Outre des méthodes expérimentales telles que l'analyse modale, nous avons établi un modèle analytique pour calculer les fréquences naturelles et les rapports d'amortissement pour des configurations de robots arbitraires. Nous avons montré que le modèle proposé prenait en compte l'effet de la gravité et les paramètres du contrôleur du robot.

Le deuxième objectif est de contribuer à une procédure d'optimisation afin d'identifier les paramètres de rigidité et d'amortissement des joints dans lesquels les informations modales incomplètes sont supposées être mesurées dans plusieurs configurations de robots. Pour les robots à articulation flexible, ces paramètres sont essentiels pour comprendre leur comportement dynamique. Nous avons montré que la procédure proposée pouvait identifier des paramètres inconnus malgré que les rapports d'amortissement soient estimés avec des écarts.

Le dernier objectif est de proposer un algorithme efficace pour résoudre le problème de la dynamique inverse dans des applications en temps réel. Le produit de cet algorithme qui comprend une liste d'expressions mathématiques écrites en code C/Matlab optimisé, peut être utilisé dans la conception de lois de contrôle avancées basées sur des modèles. Nous avons démontré que notre algorithme pouvait être utilisé pour contrôler des robots articulaires flexibles avec les lois de contrôle d'action prédictive.

Mots-clés: robot à joint flexible, linéarisation, analyse modale, identification, dynamique inverse, différenciation symbolique,

Dynamic Modeling and Modal Analysis of Multiple Degrees of Freedom Robot Manipulators with Flexible Joints

Thanh Trung DO

ABSTRACT

Thanks to of their production flexibility and safety in human interactions, robots with rigid links and flexible (elastic) joints have recently received the attention from researchers and application engineers. In the past, the elasticity of the joints was often neglected due to the complexity of the robot dynamic models. In addition, the limitations in the ability to measure and calculate the dynamics led to the robot model with rigid joints being often used for simulation and control purposes. However, this approach can lead to the robot instability during its functioning.

The first objective of this research is to develop relevant mathematical models for flexible joint robots in order to simulate and predict their dynamic behavior. In addition to experimental methods such as modal analysis, we established an analytical model to compute the natural frequencies and damping ratios for arbitrary robot configurations. We showed that the proposed model could consider the effect of gravity and the parameters of the robot's controller.

The second objective is to contribute an optimization procedure to identify the stiffness and damping parameters of joints in which the incomplete modal information is assumed to be measured in several robot configurations. For flexible joint robots, these parameters are essential to understanding their dynamic behavior. We showed that the proposed procedure could identify unknown parameters, even when the damping ratios are estimated with deviations.

The last objective is to propose an efficient algorithm to solve the inverse dynamics problem in real-time applications. The output of this algorithm, including a list of mathematical expressions written in optimized C/Matlab code, can be used in the design of advanced model-based control laws. We demonstrated that our algorithm can be used to control flexible joint robots with the feedforward control laws.

Keywords: flexible-joint robot, linearization, modal analysis, identification, inverse dynamics, symbolic differentiation,

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EMA	Experimental Modal Analysis
OMA	Operational Modal Analysis
AMA	Analytical Modal Analysis
FRF	Frequency Response Function
AR	Autoregression Model
VAR	Vector Autoregression Model
TFB	Transmissibility Function-Based Method
MAC	Modal Assurance Criterion
OMAC	Order Modal Assurance Criterion
NAFID	Natural Frequency Identification
SVD	Singular Value Decomposition
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
DH	Denavit-Hartenberg
BF	Body-Fixed Frame
VSA	Variable Stiffness Actuators
SEA	Series Elastic Actuators
NOC	Natural Orthogonal Complement
RNEA	Recursive Newton-Euler Algorithm

EJNEA	Elastic Joint Recursive Newton-Euler Algorithm
RLGA	Recursive Algorithm Using Lie Group
SRNEA	Symbolic Recursive Newton-Euler Algorithm
DRNEA	Derivative of Recursive Newton-Euler Algorithm
SDRNEA	Symbolic Derivative of Recursive Newton-Euler Algorithm

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\boldsymbol{q}_{\mathrm{m}}$	Vector of motor coordinate (rad)
$\boldsymbol{q}$	Vector of joint coordinates (rad or m)
$\boldsymbol{M}$	Mass matrix of rigid links
$\boldsymbol{B}$	Mass matrix of rotors ($\mathrm{kg \cdot m^2}$)
$\boldsymbol{C}$	Coriolis matrix
$\boldsymbol{g}$	Vector of gravity
$\boldsymbol{K}$	Stiffness matrix ($\mathrm{N \cdot m/rad}$)
$\boldsymbol{D}$	Damping matrix of joints ($\mathrm{N \cdot m \cdot s/rad}$ or $\mathrm{N \cdot s/m}$)
$\boldsymbol{D}_{\mathrm{m}}$	Damping matrix of motors ($\mathrm{N \cdot m \cdot s/rad}$).
$\boldsymbol{W}$	Matrix of characteristic radius of the joint
$\boldsymbol{D}_{\mathrm{m}}$	Damping matrix of motors ($\mathrm{N \cdot m \cdot s/rad}$)
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{K}_{\mathrm{P}}$	Proportional gain matrix
$\boldsymbol{K}_{\mathrm{D}}$	Derivative gain matrix
$\boldsymbol{\tau}_{\mathrm{ext}}$	Vector of external forces/moments ($\mathrm{N \cdot m}$ or N)
$\boldsymbol{\tau}_{\mathrm{m}}$	Vector of motor torques ($\mathrm{N \cdot m}$)
$\boldsymbol{\tau}_{\mathrm{f}}$	Vector of joint friction torques ($\mathrm{N \cdot m \cdot s/rad}$ or $\mathrm{N \cdot s/m}$)
$\boldsymbol{\tau}_{\mathrm{f,m}}$	Vector of motor friction torques ($\mathrm{N \cdot m \cdot s/rad}$)
$\boldsymbol{A}$	State matrix

λ	Eigenvalue
$\boldsymbol{y}$	Eigenvectors
ω	Natural frequency (rad)
ζ	damping ratio (-)
$\boldsymbol{I}$	Identity matrix
$\overline{\boldsymbol{M}}$	Linearized mass matrix
$\overline{\boldsymbol{D}}$	Linearized damping matrix
$\overline{\boldsymbol{K}}$	Linearized stiffness matrix
$\overline{\boldsymbol{S}}$	Matrix square root
$\boldsymbol{k}$	Vector of stiffness parameters (N·m/rad or N/m)
$\boldsymbol{d}$	Vector of damping parameters (N·m·s/rad or N·s/m)

INTRODUCTION

0.1 Motivation

Robot manipulators are defined as rigid bodies interconnected by joints. One of the most advantageous features of robots is that they can be reprogrammed to perform complicated tasks in a large workspace.

Industrial requirements are steadily increasing with regards to the accuracy and load-carrying capabilities of robots, while the flexibility of the robot's components, such as links, joints, gear/belt transmissions, has become increasingly significant. The accuracy of the robot's end-effector is limited by its static and dynamic deflections, especially when the robot is controlled at high speed for assembling tasks or when it is used in a machining process that requires high contact forces with the workpiece. In general, the deflections of manipulators are dependent upon the following factors (Dwivedy & Eberhard, 2006; De Luca & Book, 2016):

- transmissions elements;
- the total lengths of links;
- link materials and shapes;
- the masses and positions of the centers of masses; and
- the contact forces/moments.

The use of special materials with high stiffness and strength in the design of the robot structure can improve robot accuracy. In addition, optimal design methods can be applied to maximize their fundamental frequency, i.e. to reduce the vibration of the links. In many applications, manipulators are designed for specific purposes, such as a larger workspace, higher speed, lower energy consumption, or lower rigidity for safer operation, in which the flexibility of the joints is of more concern than the flexibility of the links.

Overall, the main problem of manipulators with flexible components is how to control their vibrations to achieve higher accuracy and avoid resonant phenomena that may cause damage to the robot's mechanical structure.

0.2 Research problems

For the case of a robot with rigid links and flexible joints, the flexible joints introduce extra degrees of freedom into the system. In addition, because the number of actuators is less than the number of degrees of freedom of the robot, the control problem of flexible joint robots is quite challenging. This research focuses on the vibration and control problems based on numerical simulations.

To improve the accuracy of the flexible joint robot used in a machining process, the vibration problem of the robot should be studied. In principle, the dynamic behavior of a robotic system for a given configuration can be characterized by modal parameters, including natural frequencies, damping ratios and mode shapes. These parameters can be identified from the input and output responses measured using the experimental modal analysis or operational modal analysis technique (Ewins, 2000; Brincker & Ventura, 2015), in the time domain or frequency domain. While this approach is very efficient, it requires more effort if the modal analysis is conducted on several robot configurations, as the modal parameters will depend on the robot configuration and the stiffness/damping parameters of joints due to their flexibility. Several works, including those of (Chu, Zhang, Chen & Sun, 2015) and (Berninger, Fuderer & Rixen, 2020), have showed that the modal parameters are also affected by the parameters of the controller. Therefore, to compute modal parameters for any configuration, it is necessary to first establish an analytical model. In this model, all robot parameters are assumed to be predetermined. Among the kinematic and dynamic parameters of a flexible joint robot, the stiffness and damping parameters at flexible joints play an essential role in the vibration analysis and design of controllers. The stiffness parameter of the joints can be identified by measuring the deformation at each joint or deflection

of the end-effector (Dumas, Caro, Garnier & Furet, 2011) using external sensors/devices such as laser trackers. The remaining damping parameters can be estimated based on the vibration analysis using experimental measurements. Meanwhile, the use of advanced model-based control laws to control flexible joint robots is a robust approach in order to reduce undesired vibrations (De Luca, 2000; De Luca & Book, 2016). For these control laws, the inverse dynamic problem of flexible joint robots must be solved online, where the most challenging task is how to compute the high-order derivatives of link dynamics. If the dynamic models of robots are established using the Lagrangian formulation, the solution of the inverse dynamics problem quickly becomes very complicated.

0.3 Objectives

This research aims to find relevant mathematical models based on numerical simulations for the vibration analysis, parameter identification, and control of industrial manipulators with rigid links and flexible joints. To achieve this goal, the following three specific objectives are considered:

- The first objective is to develop an analytical model for computing natural frequencies and damping ratios of robots with flexible joints. The proposed model is derived from the closed-loop control system. Therefore, the effects of gravity, external forces, and control parameters are considered. Based on this model and the given physical parameters of the robot, we can simulate natural frequencies and damping ratios in the workspace.
- The second objective is to identify flexible joints' stiffness and damping parameters based on incomplete modal information measured using the modal analysis techniques. An optimization procedure is proposed in which the objective function is derived from the linearized model. These identified parameters are crucial for the model-based control problem and the simulation of the dynamic behavior of robots.

- The third objective is to develop a symbolic algorithm to generate the inverse dynamics of flexible joint robots for advanced model-based control laws. The output of this algorithm is written in the C/Matlab code and it can be used for real-time applications.

0.4 Thesis organization and outline

This article-based thesis contains four chapters and is structured as follows.

Chapter 1 presents primary the references related to three subjects: the modeling, identification, and control of flexible joint manipulators.

Chapter 2 contains the first journal article. General dynamic equations of flexible joint robots, including prismatic and/or revolute joints, are established based on the Lagrangian formulation. The effect of robot's tool and external forces/moments applied on the end-effector is also considered in this model. From the dynamic model, linearized dynamic equations are developed for the vibration analysis. The effect of control parameters is also included in the proposed linearized model. Therefore, modal parameters for any robot configuration can be computed straightforwardly. Modal parameters obtained from the linearized model are validated using the operational modal technique based on the vector autoregression model.

Chapter 3 presents the second journal article. The purpose of this chapter is to discuss a new optimization procedure to identify the stiffness and damping parameters of flexible joint robots. The proposed method is developed based on the inverse eigenvalue problem for linear systems and the model updating in structural dynamics. From natural frequencies and damping ratios simulated/measured (using simulated/experimental data) and the linearized dynamic model obtained in chapter 2, stiffness and damping parameters are found by solving the least-square optimization problem. The influence of damping ratio deviation on identification parameters is also investigated.

Chapter 4 presents the third journal article for the model-based control problem. A new symbolic differentiation algorithm is proposed to automatically generate the inverse dynamics of flexible joint robots in optimized code. The proposed algorithm is developed based on the well-known recursive Newton-Euler algorithm for rigid bodies, recursive differentiation and computer algebra systems such as Maple software. The results obtained in this chapter can be used for real-time simulation and control purposes, e.g., a model-based control design using feedforward + PD control. The proposed algorithm outperforms other algorithms.

Conclusions are presented at the end of the thesis, along with several recommendations for future work.

0.5 Contributions

The following papers with reviewing committees have been submitted/published in international journals and conferences by the supervisors and author during the completion of this research.

Journal papers

1. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Linearization of dynamic equations for vibration and modal analysis of flexible joint manipulators*. Published in *Mechanism and Machine Theory*, vol. 167, January 2022.
2. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Identification of stiffness and damping parameters for flexible joint manipulators*. Submitted in *Acta Mechanica*, November 2021.
3. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Symbolic differentiation algorithm for inverse dynamics of serial robots with flexible joints*. Published in ASME-*Journal of Mechanisms and Robotics*, vol. 13, no. 6, December 2021.

Conference papers

1. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *NAFID-A Grid Tool for output only modal analysis*. Published in the Surveillance conference, INSA Lyon, France, pp. 119-127, 2019.
2. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Analytical modal analysis of flexible joint manipulators*. Published in the 27th Canadian Congress of Applied Mechanics Conference (CANCAM2019), Sherbrooke, Québec, Canada, 2019.
3. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Automatic generation of inverse dynamics for industrial robots with flexible joints using a computer algebra*-Poster. Published in the conference: Applications of Computer Algebra (ACA2019), Montréal, Québec, Canada, 2019.
4. Thanh-Trung Do, Viet-Hung Vu, Zhaoheng Liu, *Dynamic linearization and modal analysis of flexible manipulators*. Published in the 27th International Congress on Sound and Vibration (ICSV27), Praha, Czech Republic, 2021.

CHAPTER 1

LITERATURE REVIEW

This chapter presents a literature review for the dynamic modeling, modal analysis, parameter identification, and inverse dynamic control problems of manipulators.

1.1 Dynamic modeling

This section offers a brief review of the dynamic models for the flexible joint robots used for model-based simulation and control purposes. The critical assumptions summarized in (De Luca & Book, 2016) are used to orient this review. Generally, a flexible joint robot can be considered as a multibody dynamics system in which its dynamic equations can be established using the Lagrangian formulation, the Newton-Euler equations, Kane dynamics, the principle of virtual power, and the principle of virtual work (Schiehlen, 1990, 1993). To describe the kinematic quantities of a robot, the standard/modified Denavit-Hartenberg notations (Tsai, 2003; Khalil & Dombre, 2004; Siciliano, Sciavicco, Villani & Oriolo, 2009) can be used.

1.1.1 Generalized coordinates

Because of the presence of flexible joints, more generalized coordinates are required to describe the configuration of a flexible joint robot than that of the robot with rigid joints. The following coordinates can be used to describe the configuration of a flexible joint i: the link (joint) coordinate q_i, the motor angle before the reduction gear $q'_{\mathrm{m,i}}$, and the motor angle after the reduction gear $q_{\mathrm{m,i}}$ (see Fig. 4.1 for more details). The relation between motor coordinates $q'_{\mathrm{m,i}}$ and $q_{\mathrm{m,i}}$ is given by: $q'_{\mathrm{m,i}} = N_i q_{\mathrm{m,i}}$ with the gear ratio $N_i \geq 1$. Therefore, vectors of coordinates

for the robot with n flexible joints can be grouped as:

$$\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^\mathsf{T} \tag{1.1}$$

$$\boldsymbol{q}'_\mathrm{m} = [q'_{\mathrm{m},1}, q'_{\mathrm{m},2}, \ldots, q'_{\mathrm{m,n}}]^\mathsf{T} \tag{1.2}$$

$$\boldsymbol{q}_\mathrm{m} = [q_{\mathrm{m},1}, q_{\mathrm{m},2}, \ldots, q_{\mathrm{m,n}}]^\mathsf{T} \tag{1.3}$$

1.1.2 Dynamic equations of a robot with multiple flexible joints

We consider a manipulator with n flexible joints controlled by n electrical motors. Based on the multibody dynamic theory, the robot is modeled as $2n$ rigid bodies including n moving rigid links and n rotors. Note that the motor stators are combined with the rigid links for simplification.

When the Lagrangian formulation is applied, the kinetic energy of all the rigid links and motors must be computed, as well as the potential energy of the rigid bodies and elastic springs due to flexible joints. Generalized forces/moments can be computed from the virtual work. The dynamic equations can also include the friction forces/moments at the motors and joints, based on the dissipation function.

The complete model

When accounting for the inertial coupling between links and rotors, the dynamic equations of the robot are expressed in the complete model as follows (Tomei, 1991; De Luca & Tomei, 1996):

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{S}(\boldsymbol{q}) \\ \boldsymbol{S}^\mathsf{T}(\boldsymbol{q}) & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}}_\mathrm{m} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) & \boldsymbol{C}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{C}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}}_\mathrm{m} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}(\boldsymbol{q}) \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{K} \\ -\boldsymbol{K} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{q}_\mathrm{m} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}_\mathrm{f}(\dot{\boldsymbol{q}}) \\ \boldsymbol{\tau}_{\mathrm{f,m}}(\dot{\boldsymbol{q}}_\mathrm{m}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}^\mathsf{T}(\boldsymbol{q})\boldsymbol{\tau}_\mathrm{ext} \\ \boldsymbol{\tau}_\mathrm{m} \end{bmatrix} \tag{1.4}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is the symmetric mass matrix of the rigid links (including the mass of motors), $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ is the diagonal inertia matrix of the rotor (multiplied by squared gear ratios), $\boldsymbol{S} \in \mathbb{R}^{n\times n}$

is the upper-triangular mass matrix due to inertial coupling between the links and the rotors ; $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is the Coriolis matrix of rigid links, $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \mathbb{R}^{n\times n}$ are the Coriolis matrices due to inertial coupling between the links and the rotors; $\boldsymbol{g} \in \mathbb{R}^n$ is the vector of gravity forces/torques, $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is the constant diagonal matrix of shaft torsional stiffness coefficients, $\boldsymbol{\tau}_{\mathrm{f}},\ \boldsymbol{\tau}_{\mathrm{f,m}} \in \mathbb{R}^n$ are the vectors of joint and motor friction torques; $\boldsymbol{\tau}_{\mathrm{m}} \in \mathbb{R}^n$ is the vector of the motor's torques; $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the geometric Jacobian matrix, and $\boldsymbol{\tau}_{\mathrm{ext}} \in \mathbb{R}^6$ is the vector of external forces/torques at the end-effector.

The reduced model

Based on the assumption originally proposed in (Spong, 1987) in which a robot's electrical motors have high gear ratios (> 100), the angular velocity of the rotor i is much greater than the angular velocity of the rigid link $i-1$, which the motor is fixed in. Therefore, the effect of inertial couplings between links and rotors can be neglected. Consequently, the equations of motion of the robot are rewritten in the reduced model as follows:

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}(\boldsymbol{q}) \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{K} \\ -\boldsymbol{K} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{q}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}_{\mathrm{f}}(\dot{\boldsymbol{q}}) \\ \boldsymbol{\tau}_{\mathrm{f,m}}(\dot{\boldsymbol{q}}_{\mathrm{m}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{ext}} \\ \boldsymbol{\tau}_{\mathrm{m}} \end{bmatrix} \quad (1.5)$$

It can be seen in the reduced form that the upper and lower sub-matrices in the generalized mass matrix have vanished, and therefore, the Coriolis sub-matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are zero.

1.2 Modal analysis

The well-known modal analysis techniques can be used to identify the modal parameters (natural frequencies, damping ratios and mode shapes) of mechanical systems, structures and robotic systems (Ewins, 2000; Brincker & Ventura, 2015). The experimental modal analysis needs the input and output responses to be measured, while the operational modal analysis only needs

to measure the output response. Another method, analytical modal analysis, can also be used for this purpose. In this approach, a linearized model must be derived appropriately from the dynamical model of the investigated system. This section presents some of the literature on identifying/computing the modal parameters of robots.

In (Vu, Liu, Thomas, Li & Hazel, 2016) the authors presented an algorithm based on the vector autoregression model to identify natural frequencies and damping ratios of a Scompi robot in the grinding process. Based on the operational modal analysis technique, the modal information was extracted from acceleration signals. The discrete modal participation factor was proposed to distinguish decoupled and coupled mode shapes in different directions. The decoupled mode shapes can be further used to identify the joint stiffness.

A method to identify the modal parameters of a lightweight robot with a rotating tool installed at the end-effector was presented in (Vu, Liu, Thomas & Hazel, 2017). Their proposed method is a combination of the vector autoregressive model and the sliding window technique, and thus. it can be used for a system with slow-variable configurations. This method was validated by numerical simulation for the system with 3-DOFs and experimentally under different working conditions, including stationary and non-stationary.

In (Mousavi, Gagnol, Bouzgarrou & Ray, 2017) the authors proposed a linearized dynamic model based on the matrix structural analysis method to compute the modal parameters of robots in which the flexibility of both joints and links were considered. The proportional damping model was used to describe the damping effect. The authors demonstrated that this model could be used to analyze and predict the dynamic behavior of the Stäubli industrial robot along a machining trajectory. Furthermore, the authors integrated the cutting force model in milling operations into the linearized model to establish a stability limit diagram for chatter prediction.

In (Sun, Zhang & Dong, 2020), a prediction model of natural frequencies for a 6R industrial robot was proposed based on the partial least square method. The main idea of this approach is the use of natural frequencies measured in several selected configurations using modal analysis techniques to construct the prediction model, based on the joint coordinates. The resulting

model can thus be used to compute the natural frequencies of robots in the whole workspace. This work does not consider the damping ratio distribution in the workspace.

1.3 Parameter identification problem

A method based on fuzzy logic combined with a genetic algorithm was proposed to identify the stiffness parameters of joints for modular robots in (Li, Liu, Peng & Liu, 2002). The damping parameters cannot be identified with this method. The objective function is defined as the sum of the squared differences between the natural frequencies obtained from the analytical and experimental models. The authors demonstrated that by converting the equivalent deterministic model into a fuzzy model, the accuracy of the joint parameters could be improved by using a genetic algorithm.

Authors in (Lightcap & Banks, 2007) used the motion capture method for link position measurements to identify all the parameters of the Mitsubishi PA10-6CE robot with rigid links and flexible joints. The regressor matrix was established based on the motor angles and link positions. Therefore, all the robot parameters, including the joint stiffness parameters, could be estimated using the least-squares method.

In (Wernholt & Moberg, 2011), nonlinear gray-box models were proposed to estimate the elasticity parameters (spring-damper pairs) of a six-axis industrial robot in which the objective function is derived by minimizing the discrepancy between the nonparametric and the parametric frequency response functions. The authors demonstrated that stiffness parameters were accurately estimated, but the damping parameters required further refinement.

In (Gautier, Jubien, Janot & Robet, 2013) researchers presented a method to identify the joint stiffness of a single flexible joint robot. Because the actual and simulated motor forces/torques are used for the identification problem, the estimation of the joint position and its derivatives is unnecessary as presented in (Lightcap & Banks, 2007). This method can be expanded for multi-flexible joint robots.

In (Bottin, Cocuzza, Comand & Doria, 2020) the joint stiffness of an Adept robot has been experimentally identified using the selective modal approach. The authors demonstrated that the modal stiffness and damping coincide with joint stiffness and damping. Therefore, the stiffness parameters of joints can be computed from the identified natural frequencies. Based on the penalty function defined as the sum of the squared differences between the measured and simulated natural frequencies, the stiffness parameters are further improved by solving the optimization problem. In addition, a formulation was proposed to compute the damping parameters from the stiffness parameters and natural frequencies.

1.4 Inverse dynamics control problem

Theoretically, the control algorithms designed for robots with rigid links and rigid joints presented in (Siciliano *et al.*, 2009; Chung, Fu & Kröger, 2016) can be used to control robots with flexible joints if the robot joint has a high enough stiffness, the effect of external forces/moments on the robot links and end-effector are small, and the desired robot motion is low. To achieve high performance, the effect of elastic joints must be considered. This section briefly reviews several control laws based on the inverse dynamics problem of the flexible joint robot.

In (De Luca, 2000), authors presented feed-forward control laws based on the inverse dynamics of flexible joint robots. The input motor torques have two parts: a feed-forward part and a feedback part. The first part relates to the nominal motor torques computed from the dynamic model for a given desired motion. To compute inverse dynamics, the desired joint trajectory should be at least four times differentiable if using the reduced model Eq. (1.5), while if using the complete model Eq. (1.4), the desired joint trajectory should be at least $2(n + 1)$ times differentiable (n is the number of flexible joints). To compensate for model uncertainties or external disturbance, the PD control is combined with the first part. The main advantage of this approach is that the control law does not require additional sensors to measure the joint positions. However, the inverse dynamics must be computed online for the real-time tracking problem with different desired trajectories. This method can also be applied for robots with both rigid and flexible joints (De Luca, 1998).

The control problems of visco-elastic joint robots were addressed in (De Luca, Farina & Lucibello, 2005). The authors state that for any value of the joint viscosity, the numerical stability of the dynamic feedback design can be obtained when using either the reduced or the complete dynamic models of robots.

In (Albu-Schäffer, Ott & Hirzinger, 2007) a unified passivity-based control was proposed for the control problems of flexible joint robots, such as position, torque and impedance control. The main advantage of passivity-based controllers is that they are robust to uncertainties parameters of the robot or load parameters. However, this approach requires additional torque sensors to measure joint torques for the state feedback, while motor positions are measured using the encoders implemented on the motors. For this approach, numerical differentiation methods are used to estimate the motor speed and the first derivative of the joint torques.

In (Moberg & Hanssen, 2008) two control laws, including feedback linearization and feedforward control for flexible joint robots, are investigated where the robot parameters and state measurements are assumed to be perfect. The authors show that feedforward control performs better than feedback linearization control when considering the effect of the sampling rate, measurement noise, and parameter uncertainty.

A new online algorithm was proposed in (Buondonno & De Luca, 2015) to compute the inverse dynamics of flexible joint robots in real-time control. This algorithm (developed based on the recursive Newton-Euler algorithm for robots with rigid links) can be used to design the feedforward control law or the feedback linearization control law. Soon after, in (Buondonno & De Luca, 2016), the authors applied this algorithm to control robots with variable stiffness actuation.

When industrial robots move in a high-speed motion for the given trapezoidal velocity profile, the effect of joint elasticity is significant. In order to eliminate vibration in this case, three input shaping methods were proposed in (Kim & Croft, 2018) in which the controller is based on the PD control on the motor side and feedforward control is based on a rigid model to control the

robots. From their simulation and experimental results on a 6DOF industrial robot, the authors found that the proposed methods could apply to real robots.

A novel global tracking control approach was presented in (Giusti, Malzahn, Tsagarakis & Althoff, 2018) to control flexible joint robots with high gear ratios and high joint stiffness. Based on the reduced dynamic model, the authors combine two control schemes: inverse dynamics control and passivity-based tracking control, to guarantee that the proposed controller is robust against model uncertainties and external disturbances. The results obtained for tracking control of a 7-DOF robot with mixed rigid/elastic joints show the effectiveness of this approach.

CHAPTER 2

LINEARIZATION OF DYNAMIC EQUATIONS FOR VIBRATION AND MODAL ANALYSIS OF FLEXIBLE JOINT MANIPULATORS

Thanh Trung Do[1] , Viet Hung Vu[1] , Zhaoheng Liu[1]

[1] Department of Mechanical Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper published in *Mechanism and Machine Theory*, vol. 167, January 2022

Abstract

This paper presents the dynamic model and analytical modal analysis for robotic manipulators with rigid links and flexible joints. Dynamic equations of general robots with both prismatic and revolute joints are firstly developed using the Lagrangian formulation in minimal joint and motor coordinates. Next, linearized dynamic equations taking into account the influence of gravity forces, external forces, and control parameters are formulated based on the Taylor series. Therefore, the robot's modal parameters can be computed for any configuration based on a state-space matrix derived from the linearization model. To illustrate the proposed method, modal parameters of a flexible joint robot with six degrees of freedom are computed using the analytical method and estimated using the operational modal technique based on the vector autoregression model. Results obtained by both methods agree very well with each other.

2.1 Introduction

The dynamic behavior of mechanical systems, such as structures, machines, and robotic systems, is characterized by modal information, including three parameters: natural frequencies, damping ratios, and mode shapes. When available, these parameters can be used to predict the system's local dynamic behavior, set up a default detection and diagnostics baseline, or update the system's stiffness and damping parameters (Friswell & Mottershead, 1995). In machining operations, the knowledge about modal parameters in the workspace is very important because they can be used

to adjust cutting parameters appropriately to improve machining accuracy, i.e., to avoid chatter vibration (Quintana & Ciurana, 2011).

In practical applications, the estimation of modal parameters is complicated because it is necessary to conduct experiments to measure physical quantities at specific points using sensors, such as position, velocity, acceleration signals and even excitation forces/moments.

The well-known experimental modal analysis technique (EMA) (Maia & Silva, 1997) can be used to identify the modal parameters of mechanical systems. This technique requires output and input responses (excitation forces/moments) for modal parameter estimation algorithms. The measured data is used to establish frequency response functions (FRFs) (Ewins, 2000). Therefore, modal parameters can be extracted from these functions. For the EMA technique, the excitation forces can be generated by impact hammers, and then both excitation forces and acceleration signals are measured simultaneously using force sensors and accelerometers.

In machining operations, the measurement of excitation forces of a machine for the modal analysis is complicated because excitation sources are affected by several factors, such as random loads and the controller of the machine. In addition, it is unfeasible to stop the working machine to measure input and output responses. The cost of force sensors is also an essential factor that should be taken into account. The use of output-only responses to identify modal parameters leads to an innovative technique called operational modal analysis (OMA) (Brincker & Ventura, 2015). For this technique, modal parameters can be extracted from output responses in the time/frequency domain.

The estimation of modal parameters for robotic systems is challenging because such systems are considered as time-varying systems. In Karim, Hitzer, Lechler & Verl (2017); Hao, Wang, Liu & Yun (2020), the author used the EMA technique to estimate modal parameters of industrial robots in their workspace. In Vu *et al.* (2016), the vector autoregression model is used to identify modal information of the lightweight Scompi robot in working operations. The dynamic behavior of a milling robot depending on its configurations was investigated in Maamar, Gagnol, Le & Sabourin (2020) in which robot's modal parameters are extracted using the OMA technique

based on the transmissibility function-based method (TFB) (Devriendt & Guillaume, 2008). Overall, modal parameters of a robot depend on its configuration (Zaghbani, Songmene & Bonev, 2013; Mejri *et al.*, 2016) in which the robot's parameters such as mass, inertia, stiffness, damping parameters are assumed to be constant. In several works such as Bernzen (1999); Chu *et al.* (2015); Berninger *et al.* (2020), the authors have pointed out that the controller's effect should be considered for estimating modal parameters.

Most works mentioned above estimate modal parameters of a robot at several points in the robot's workspace. The main reason is that the use of the EMA technique, for example, to identify modal parameters, is costly and time-consuming. Generally, it is impossible to conduct modal tests at every robot's configuration in its workspace.

Recently, several authors have developed analytical methods to compute modal parameters of flexible multibody and robotic systems (Palomba & Vidoni, 2019; Bottin *et al.*, 2020). These methods have been developed based on linearized equations (Ginsberg, 1998; Lynch & Vanderploeg, 1995). The most advantage of this approach is that modal parameters can be computed in the whole workspace. However, analytical methods require knowledge about the robot's kinematic/dynamic parameters.

A new analytical formulation is proposed to compute modal parameters of flexible-joint manipulators in which the influence of gravity forces, external forces and PD control parameters is taken into account. The robot's full modal mapping can be obtained in a single computation run when the kinematic/dynamic parameters are predetermined. This modal mapping is beneficial because it allows us to quickly extract the robot's modal information at the desired configuration. For the validation purpose, the OMA technique based on the vector autoregression (VAR) model is used to estimate modal parameters from motor positions instead of from acceleration signals as the traditional approach.

The rest of the paper is organized as follows. In Section 2.2, the dynamical model of flexible joint robots is presented using the Lagrangian formulation. Next, the PD controller with the gravity compensation is discussed in order to establish the closed-loop dynamics for analytical modal

analysis. In Section 2.3, the state-space matrix is formulated to compute modal parameters. In Section 2.4, frequencies and damping ratios of a 6-DOF Scompi flexible joint robot with both prismatic/revolute joints are computed. The OMA technique based on the VAR model is used to re-estimated modal parameters obtained from the analytical model. Finally, conclusions are given in Section 2.5.

2.2 Dynamics of flexible-joint robots

2.2.1 Assumptions

In order to derive equations of motion of a general robot with n flexible joints, the following assumptions adapted from De Luca & Book (2016) are used in this work:

1. All links are considered as rigid bodies, in which the link $(i+1)$ is controlled by the motor i fixed to the link i.
2. All flexible joints are considered as lumped-parameter models.
3. Gear ratios of the motors are large, (> 100).

By using the last assumption (Spong, 1987), equations of motion of the robot can be written in reduced form. Note that this form is benefit for the model-based control (Buondonno & De Luca, 2015).

2.2.2 Dynamic equations

Based on the Lagrangian formulation (Siciliano *et al.*, 2009), equations of motion of a general flexible-joint robot are expressed in a matrix form as follows:

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}(\boldsymbol{q}) \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{m}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}\boldsymbol{W}^2 & -\boldsymbol{K}\boldsymbol{W} \\ -\boldsymbol{K}\boldsymbol{W} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{q}_{\mathrm{m}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{ext}} \\ \boldsymbol{\tau}_{\mathrm{m}} \end{bmatrix} \tag{2.1}$$

in which all terms in Eq. (2.1) are explained in Appendix I.1.

In the literature (De Luca & Tomei, 1996; Ott, 2008; De Luca & Book, 2016), authors considered flexible-joint robots with revolute joints only. By using the matrix $\boldsymbol{W}$ proposed in this work, dynamic equations of flexible-joint robots are derived in a general form in which both prismatic and revolute joints are considered.

2.2.3 Closed-loops dynamics

In order to study the dynamic behavior of a flexible joint robot, controllers should be used to track/follow the reference paths or points in the joint/task space. These controllers are used to solve two main control problems: tracking control and set-point control, in which the vector of motor torques $\boldsymbol{\tau}_{\mathrm{m}}$ is often formulated based on feedback signals measured using sensors implemented in the robot. In De Luca (2000); De Luca & Book (2016), several control strategies were proposed to compute $\boldsymbol{\tau}_{\mathrm{m}}$, such as the simple PID control, the feedforward+PD control, and computed-torque control.

For the modal analysis problem of a flexible joint robot in its workspace, the simple PD control scheme with gravity compensation proposed in Tomei (1991) is applied for the set-point control. This controller is the special case of the model-based feedforward+PD method (De Luca, 2000). Based on this control law, the desired (reference) position of joints is: $\boldsymbol{q}^{\mathrm{d}} = \text{const}$, while their velocity and acceleration are: $\dot{\boldsymbol{q}}^{\mathrm{d}} = \boldsymbol{0}$ and $\ddot{\boldsymbol{q}}^{\mathrm{d}} = \boldsymbol{0}$. Therefore, the motor torque vector is formulated as follows:

$$\boldsymbol{\tau}_{\mathrm{m}} = \underbrace{\boldsymbol{W}^{-1}\left(\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) - \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q}^{\mathrm{d}})\boldsymbol{\tau}_{\mathrm{ext}}\right)}_{\text{feedforward}} + \underbrace{\boldsymbol{K}_{\mathrm{P}}(\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} - \boldsymbol{q}_{\mathrm{m}}) - \boldsymbol{K}_{\mathrm{D}}\dot{\boldsymbol{q}}_{\mathrm{m}}}_{\text{PD controller}} \tag{2.2}$$

where $\boldsymbol{K}_{\mathrm{P}}$ and $\boldsymbol{K}_{\mathrm{D}}$ are proportional and derivative gain matrices, respectively. These matrices can be chosen as diagonal positive definite matrices to guarantee the system stability. In Eq. (2.2), $\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}}$ is the desired position of motors (rotors) which is determined from the first line of Eq. (2.1)

by setting $\ddot{\boldsymbol{q}}^{\mathrm{d}} = \mathbf{0}$ and $\dot{\boldsymbol{q}}^{\mathrm{d}} = \mathbf{0}$. Thus, one obtains:

$$\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} = \boldsymbol{W}\boldsymbol{q}^{\mathrm{d}} + (\boldsymbol{K}\boldsymbol{W})^{-1}\left(\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) - \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q}^{\mathrm{d}})\boldsymbol{\tau}_{\mathrm{ext}}\right) \tag{2.3}$$

Substituting Eq. (2.2) into Eq. (2.1) leads to $2n$ closed-loop dynamic equations as follows:

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{W}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) - \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{ext}} \\ \boldsymbol{B}\ddot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) - \boldsymbol{W}^{-1}\left(\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) - \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q}^{\mathrm{d}})\boldsymbol{\tau}_{\mathrm{ext}}\right) - \boldsymbol{K}_{\mathrm{P}}(\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} - \boldsymbol{q}_{\mathrm{m}}) + \boldsymbol{K}_{\mathrm{D}}\dot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} = \mathbf{0} \tag{2.4}$$

2.3 Analytical modal analysis

2.3.1 Linearization

By using the following definitions:

$$z \triangleq [\boldsymbol{q}^{\mathsf{T}},\, \boldsymbol{q}_{\mathrm{m}}^{\mathsf{T}}]^{\mathsf{T}},\ \dot{z} \triangleq [\dot{\boldsymbol{q}}^{\mathsf{T}},\, \dot{\boldsymbol{q}}_{\mathrm{m}}^{\mathsf{T}}]^{\mathsf{T}},\quad \text{and}\quad \ddot{z} \triangleq [\ddot{\boldsymbol{q}}^{\mathsf{T}},\, \ddot{\boldsymbol{q}}_{\mathrm{m}}^{\mathsf{T}}]^{\mathsf{T}} \tag{2.5}$$

Eq. (2.4) can be rewritten in a compact form as:

$$\boldsymbol{\Gamma}(\ddot{z}, \dot{z}, z) = \mathbf{0} \tag{2.6}$$

The linearized dynamic equations of a flexible joint robot can be derived using the Taylor series expansion (Ginsberg, 1998; Lynch & Vanderploeg, 1995) about an 'equilibrium position' $\{\ddot{z}^{\mathrm{d}}, \dot{z}^{\mathrm{d}}, z^{\mathrm{d}}\}$ defined by:

$$\ddot{z}^{\mathrm{d}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix},\ \dot{z}^{\mathrm{d}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix},\ z^{\mathrm{d}} = \begin{bmatrix} \boldsymbol{q}^{\mathrm{d}} \\ \boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} \end{bmatrix} \tag{2.7}$$

Consequently, the linearized equations can be expressed as follows:

$$\overline{\boldsymbol{M}}\Delta\ddot{z} + \overline{\boldsymbol{D}}\Delta\dot{z} + \overline{\boldsymbol{K}}\Delta z = \mathbf{0} \tag{2.8}$$

where $\Delta\ddot{z} = \ddot{z} - \ddot{z}^{\mathrm{d}}$, $\Delta\dot{z} = \dot{z} - \dot{z}^{\mathrm{d}}$ and $\Delta z = z - z^{\mathrm{d}}$ represent the perturbation vectors about the desired equilibrium position $\{\ddot{z}^{\mathrm{d}}, \dot{z}^{\mathrm{d}}, z^{\mathrm{d}}\}$. In addition, three matrices ($\overline{\boldsymbol{M}}, \overline{\boldsymbol{D}}$, and $\overline{\boldsymbol{K}} \in \mathbb{R}^{2n\times 2n}$) in Eq. (2.8) are determined by:

$$\overline{\boldsymbol{M}} \triangleq \left.\frac{\partial \boldsymbol{\Gamma}}{\partial \ddot{\boldsymbol{z}}}\right|_{(\ddot{z}^{\mathrm{d}}, \dot{z}^{\mathrm{d}}, z^{\mathrm{d}})} = \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}^{\mathrm{d}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}, \tag{2.9}$$

$$\overline{\boldsymbol{D}} \triangleq \left.\frac{\partial \boldsymbol{\Gamma}}{\partial \dot{\boldsymbol{z}}}\right|_{(\ddot{z}^{\mathrm{d}}, \dot{z}^{\mathrm{d}}, z^{\mathrm{d}})} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{m}} + \boldsymbol{K}_{\mathrm{D}} \end{bmatrix}, \tag{2.10}$$

$$\overline{\boldsymbol{K}} \triangleq \left.\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{z}}\right|_{(\ddot{z}^{\mathrm{d}}, \dot{z}^{\mathrm{d}}, z^{\mathrm{d}})} = \begin{bmatrix} \boldsymbol{K}_{\mathrm{G}}(\boldsymbol{q}^{\mathrm{d}}) + \boldsymbol{K}\boldsymbol{W}^2 & -\boldsymbol{K}\boldsymbol{W} \\ -\boldsymbol{K}\boldsymbol{W} & \boldsymbol{K} + \boldsymbol{K}_{\mathrm{P}} \end{bmatrix} \tag{2.11}$$

where the matrix $\boldsymbol{K}_{\mathrm{G}} \in \mathbb{R}^{n\times n}$ is derived as follows:

$$\boldsymbol{K}_{\mathrm{G}}(\boldsymbol{q}^{\mathrm{d}}) = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^{\mathrm{d}}} - \left.\frac{\partial (\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{ext}})}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}^{\mathrm{d}}} \tag{2.12}$$

It can be seen that two matrices $\overline{\boldsymbol{M}}$ and $\overline{\boldsymbol{K}}$ depend on the robot's configuration, i.e., the vector of joint coordinates $\boldsymbol{q}^{\mathrm{d}}$. Furthermore, the matrices $\overline{\boldsymbol{D}}$ and $\overline{\boldsymbol{K}}$ are functions of damping and stiffness coefficients at joints/motors, as well as the stiffness and damping gains of the controller. The effect of the gravity forces and external forces/moments on the stiffness system is also included in Eq. (2.12) that makes more/less 'artificial stiffness' into the stiffness matrix $\overline{\boldsymbol{K}}$. Note that the effect of the controller on the modal parameters was also discussed in Inman (2017) for linear systems, in which the positive position feedback (PPF) control used to modify dynamic behavior of linear systems is similar to the PD control law presented above.

To the best of our knowledge, the linearized model of a general robot with both flexible revolute/prismatic joints presented in Eq. (2.8) including the effects of gravity and external forces/moments Eq. (2.12) is firstly proposed in this study.

2.3.2 Modal parameters in the workspace

By assuming that the physical robot's parameters (masses, positions of center of masses, inertia tensor, and stiffness/damping parameters), as well as the gain matrices $\boldsymbol{K}_{\mathrm{P}}$ and $\boldsymbol{K}_{\mathrm{D}}$ of the PD controller are constants and predetermined, Eq. (2.8) can be used for the analytical modal analysis (AMA) of the robot. Because the damping matrix $\overline{\boldsymbol{D}}$ is non-proportional, modal parameters can be found by solving the following eigenvalue problem (Inman, 2017):

$$(\boldsymbol{A} - \lambda_k \boldsymbol{I})\boldsymbol{y}_k = \boldsymbol{0} \tag{2.13}$$

where the state matrix $\boldsymbol{A} \in \mathbb{R}^{4n\times 4n}$ is derived from the linearized mass, damping and stiffness matrices, ($\overline{\boldsymbol{M}}, \overline{\boldsymbol{D}}$ and $\overline{\boldsymbol{K}}$) as follows:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\overline{\boldsymbol{M}}^{-1}\overline{\boldsymbol{K}} & -\overline{\boldsymbol{M}}^{-1}\overline{\boldsymbol{D}} \end{bmatrix} \tag{2.14}$$

where $\boldsymbol{I}$ denotes the identity matrix. If a eigenvalue λ_k is found from the characteristic polynomial, $\det(\boldsymbol{A} - \lambda_k \boldsymbol{I}) = 0$, a corresponding eigenvector $\boldsymbol{y}_k \in \mathbb{C}^{4n}$ can be therefore obtained from Eq. (2.13). In practice, λ_k may be real and/or a pair of complex conjugate numbers. Therefore, three different cases may occur (Troger & Steindl, 1991):

- Case 1: If all eigenvalues have negative real parts, i.e., lie on the left-hand side of the complex plane, then the dynamic behavior of the robot at $\boldsymbol{q}^{\mathrm{d}}$ is asymptotically stable.
- Case 2: If at least one of eigenvalues has a positive real part, i.e., lies on the right-hand side of the complex plane, then the dynamic behavior of the robot at $\boldsymbol{q}^{\mathrm{d}}$ is unstable. Therefore, gain matrices need to be changed appropriately to achieve stability.
- Case 3: If one of eigenvalues has zero real part, then dynamic behavior of the robot at $\boldsymbol{q}^{\mathrm{d}}$ may be stable or unstable.

For the under-damped case, all eigenvalues are complex numbers. A complex conjugate pair can be expressed as:

$$\lambda_{k,k^*} = -\omega_k \zeta_k \pm j\omega_k \sqrt{1 - \zeta_k^2} \tag{2.15}$$

for $k = 1, \ldots, 2n$. As a result, a natural frequency ω_k and a damping ratio ζ_k are computed from λ_k as follows:

$$\omega_k = \sqrt{\text{real}(\lambda_k)^2 + \text{imag}(\lambda_k)^2} \quad \text{(rad/s)}, \tag{2.16}$$

$$\zeta_k = -\frac{\text{real}(\lambda_k)}{\omega_k} \tag{2.17}$$

where $f_k = \frac{\omega_k}{2\pi}$ (Hz). In addition, a complex mode shape vector $\boldsymbol{u}_k \in \mathbb{C}^{2n}$ corresponding to ω_k and ζ_k is determined from the first $2n$ elements of the eigenvector $\boldsymbol{y}_k$ (Inman, 2017):

$$\boldsymbol{u}_k = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \boldsymbol{y}_k \tag{2.18}$$

As a result, modal parameters of a flexible joint robot (for given the desired positions of joint coordinates $\boldsymbol{q}^{\text{d}}$, all kinematic/dynamic parameters and the PD controller gains) are computed using the following steps:

- Step 1: Evaluate the mass matrix $\boldsymbol{M}(\boldsymbol{q}^{\text{d}})$ and the matrix $\boldsymbol{K}_{\text{G}}(\boldsymbol{q}^{\text{d}})$ using Eq. (2.12).
- Step 2: Compute the linearized mass, damping and stiffness matrices using Eqs. (2.9−2.11).
- Step 3: Establish the state matrix using Eq. (2.14).
- Step 4: Solve eigenvalue problem using Eq. (2.13) in order to find eigenvalues λ_k and eigenvectors $\boldsymbol{y}_k$ for $k = 1, \ldots, 2n$.
- Step 5: Extract natural frequencies, damping ratios and mode shapes using Eqs. (2.16−2.18).

Note that in the 'Step 4', the eig function in Matlab can be used to solve the eigenvalue problem using the numerical method.

2.4 Numerical simulation

2.4.1 Scompi robot and its parameters

The lightweight Scompi robot with six degrees of freedom developed at Hydro Quebec research institute is shown in Fig. 2.1. The robot is designed to be quickly installed on a track and to

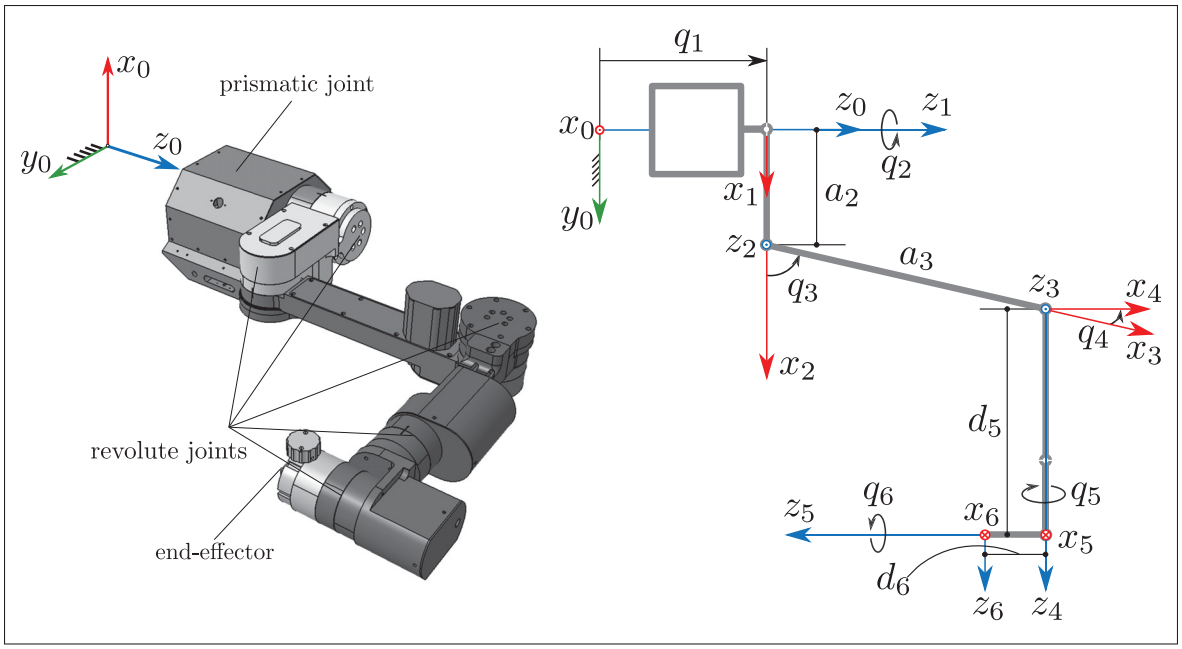

Figure 2.1 Model of the Scompi robot (left) and its kinematic representation using the standard Denavit-Hartenberg convention (right)

perform in the confined space for machining tasks of large complex hydro-power equipment such as grinding, polishing and welding (Hazel, Côté, Laroche & Mongenot, 2012).

The Scompi robot has a total weight about 39 (kg) including six rigid links connected together by six joints, in which the first joint is a prismatic joint, and the others are revolute joints. All joints are driven by stepping motors coupled to harmonic drivers with high gear ratios. The first joint is designed based on the rack-and-pinion mechanism in which r_1 is the rack gear's radius (Hazel *et al.*, 2012) defined by $r_1 = 0.057$ (m) (Lessard, Swiatek, Liu & Hazel, 2011). For other revolute joints, $r_i = 1$, for $i = 2, \ldots, 6$. In addition, the stiffness and damping parameters of the Scompi robot (Pham, Hamelin, Hazel & Liu, 2020) are presented in Table 2.1.

For the kinematic modeling of the Scompi robot, the standard Denavit-Hartenberg (DH) convention is used (Denavit & Hartenberg, 1955; Tsai, 2003) in which the DH parameters are listed in Table 2.2.

Table 2.1 Damping/stiffness parameters of joints and motors

i	1	2	3	4	5	6	**scale factor**	**unit**
$d_{\mathrm{f},i}$	2.73	24.74	5.76	9.37	8.74	8.74	1	(N·s/m) or (N·m·s/rad)
$d_{\mathrm{fm},i}$	9.46	9.46	9.46	9.46	5.90	5.90	10^{-4}	(N·m·s/rad)
k_i	25	120	120	57	29	29	1000	(N·m/rad)

Table 2.2 DH parameters of the Scompi robot with $a_2 = 0.192$, $a_3 = 0.420$, $d_5 = 0.380$, and $d_6 = 0.088$

link	1	2	3	4	5	6	**unit**
θ_i	$\frac{\pi}{2}$	q_2	q_3	q_4	q_5	q_6	(rad)
d_i	q_1	0	0	0	d_5	d_6	(m)
a_i	0	a_2	a_3	0	0	0	(m)
α_i	0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	(rad)

The other dynamic parameters including masses, inertia tensors, positions of the center of masses, gear ratios obtained from manufacturers or CAD data are presented in Table 2.3 and Table 2.4.

Table 2.3 Physical parameters of rigid links

	link 1	**link 2**	**link 3**	**link 4**	**link 5**	**link 6**	**unit**
m_i	19.0000	5.7230	5.1660	4.4250	3.3020	1.0220	(kg)
$x_{\mathrm{C},i}$	0.0210	-0.0540	-0.1380	0.0110	0.0	-0.0040	(m)
$y_{\mathrm{C},i}$	0.0310	0.0030	0.0	0.0120	-0.0260	0.0390	(m)
$z_{\mathrm{C},i}$	-0.2260	0.0140	-0.0190	0.1610	-0.0220	0.0	(m)
$I_{\mathrm{xx},i}$	0.0	0.0100	0.0080	0.0440	0.0120	0.0010	(kg·m^2)
$I_{\mathrm{xy},i}$	0.0	0.0030	0.0	0.0	0.0	0.0	(kg·m^2)
$I_{\mathrm{xz},i}$	0.0	-0.0030	-0.0100	0.0020	0.0	0.0	(kg·m^2)
$I_{\mathrm{yz},i}$	0.0	0.0440	0.1600	0.0440	0.0060	0.0010	(kg·m^2)
$I_{\mathrm{zy},i}$	0.0	0.0	0.0	0.0070	-0.0010	0.0	(kg·m^2)
$I_{\mathrm{zz},i}$	0.0	0.0430	0.1600	0.0080	0.0080	0.0120	(kg·m^2)

2.4.2 Simulation results

Using the analytical formulation presented in Section 2.3, modal parameters of the Scompi robot are computed in two cases:

Table 2.4 Gear ratios and inertia moments of motors

i	1	2	3	4	5	6	**scale factor**	**unit**
N_i	51	160	160	160	160	160	1	(-)
$I_{r_i,z}$	1.71	2.65	2.42	1.26	0.91	0.86	10^{-4}	(kg·m^2)

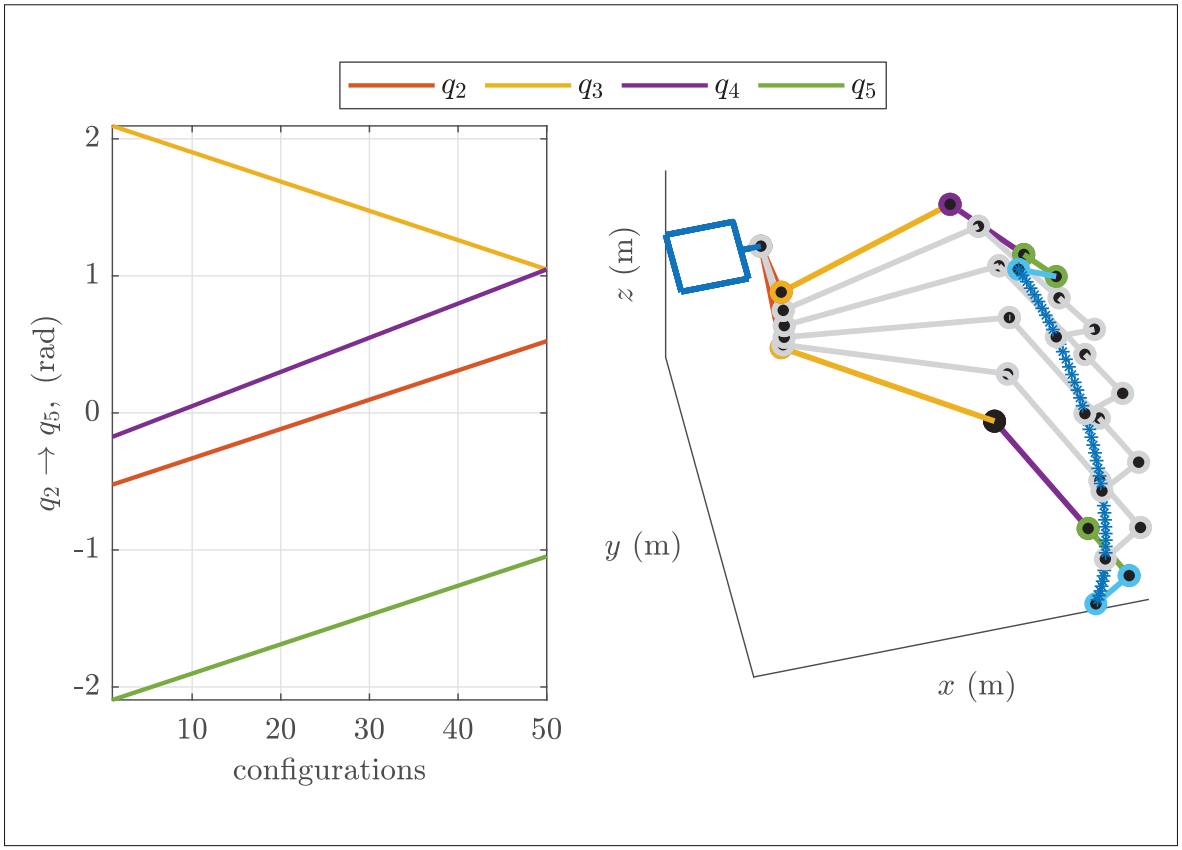

Figure 2.2 Different configurations of the robot in its workspace for computing modal parameters: robot's joints depending on configurations (left), robot' configurations in the Cartesian space (right)

Case 1: There is no robot's tool.

Case 2: A tool with the mass $m_{\mathrm{E}} = 4$ (kg) is added to the end-effector.

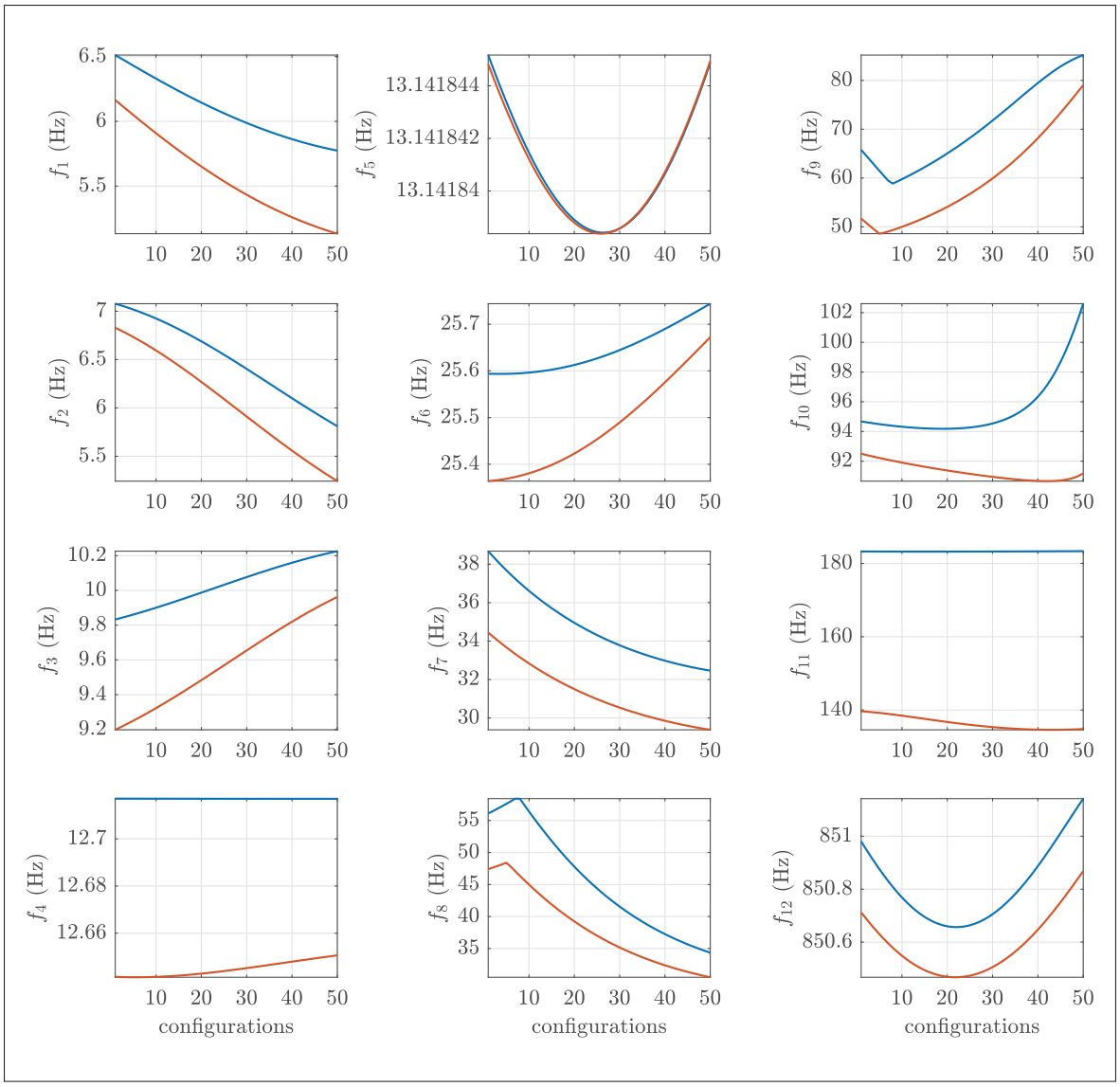

Figure 2.3 Undamped natural frequencies for 50 configurations: without additional mass (blue line), with additional mass (orange line)

To illustrate modal parameters of the Scompi robot in its workspace, 50 joint's configurations shown in Fig. 2.2 are used in which several configurations are plotted for illustration purpose. The first and last joints are hold constant with $q_1^{\mathrm{d}} = 0\,(\mathrm{m})$ and $q_6^{\mathrm{d}} = 0\,(\mathrm{rad})$, and other joints are

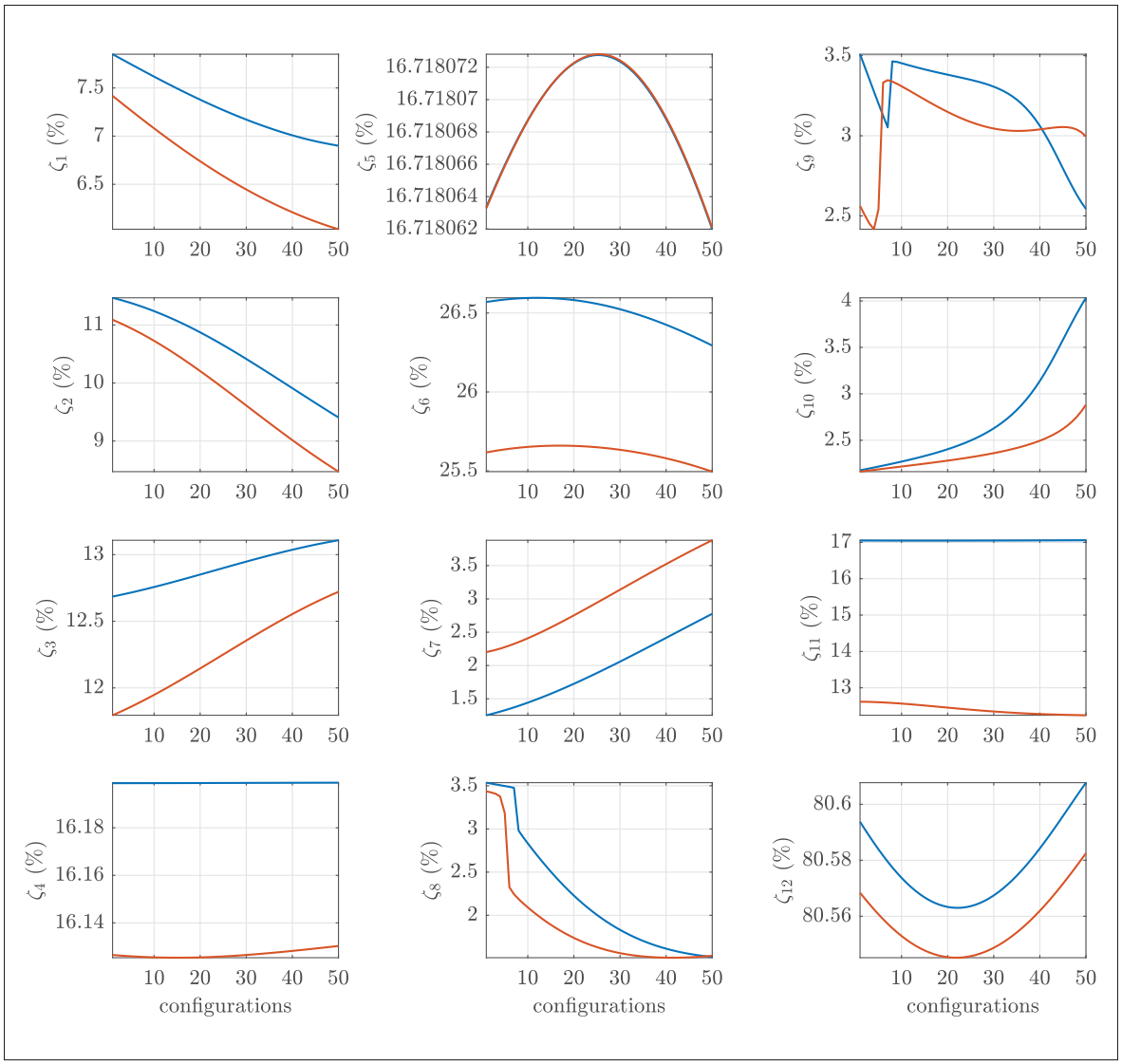

Figure 2.4 Damping ratios for 50 configurations: without additional mass (blue line), with additional mass (orange line)

varied linearly in the following intervals:

$$q_2^{\mathrm{d}} \in [-\frac{\pi}{6}, \frac{\pi}{6}],\ q_3^{\mathrm{d}} \in [\frac{2\pi}{3}, \frac{\pi}{3}],\ q_4^{\mathrm{d}} \in [-\frac{\pi}{18}, \frac{\pi}{3}],\ q_5^{\mathrm{d}} \in [-\frac{2\pi}{3}, -\frac{\pi}{3}]$$

For simplification, external forces/moments are set to zero, $\boldsymbol{\tau}_{\text{ext}} = \mathbf{0}$. In addition, the inertia tensor and position of the center of mass for the robot's tool are set to zero, i.e., $\boldsymbol{I}_{\text{E}}^{(n)} = \mathbf{0}$ and $\boldsymbol{s}_{\text{E}}^{(n)} = \mathbf{0}$ with $n = 6$.

In addition, the gain matrices $\boldsymbol{K}_{\text{P}}$ and $\boldsymbol{K}_{\text{D}}$ in Eqs. (2.9) and (2.10)) for both cases are:

$$\boldsymbol{K}_{\text{P}} = 1000\,\text{diag}(15, 15, 15, 15, 15, 15) \tag{2.19}$$

$$\boldsymbol{K}_{\text{D}} = \text{diag}(52, 52, 52, 52, 52, 52) \tag{2.20}$$

Modal parameters of twelve modes are computed simultaneously using Matlab 2018b, in which the computation time for one configuration is approximately 0.003 (s) using a Laptop with 12Gb memory, Intel Core i7, CPU @2.20 GHz. The simulation results for 50 configurations are shown in Fig. 2.3 and Fig. 2.4.

It can be seen in Fig. 2.3 that the lowest frequency is about 5.13 (Hz), while the highest frequency is 851.14 (Hz). Because the robot's tool is added to the end-effector, the natural frequency curves for the second case are clearly lower than those for the first case. In addition, the frequencies for modes 8, 9, 10, and mode 11 change significantly when adding the extra mass to the end-effector in which the deviations are approximately 12.08 (Hz), 11.70 11.60 (Hz), and 49 (Hz). In modes 4, 5, 6, and mode 12, their frequencies are nearly constant, i.e., they are independent to the robot's configuration. The modes 4 and 5 are closely spaced modes (Reynders, Houbrechts & De Roeck, 2012) because their frequencies are close the common mean, approximately 12.88 (Hz).

On the other hand, the damping ratios presented in Fig. 2.4 vary in the range: from 1.25 (%) for mode 7 to 80.61 (%) for mode 12. In practice, modes with high damping ratios are often ignored or are difficult to observe because the system does not vibrate or vibrate in a very short time. Generally, the damping ratio curves for the first case are lower than those for the second case except for mode 7. Most damping ratios for all modes are less than 27 (%). In the last mode, the damping ratios are greater than 80 (%) for both cases. It can be seen that damping ratios in

modes 4 and 5 in two cases are nearly constant, approximately 16.16 and 16.71 (%). Besides, modes 4, 5 and 12 are nearly independent of the robot's configuration.

Obviously, natural frequencies and damping ratios of the Scompi robot change very complicatedly in its workspace, and they depend clearly on the robot's configurations. Therefore, the linearized model proposed is advantageous in achieving the configuration-dependent modal parameters for flexible joint robots.

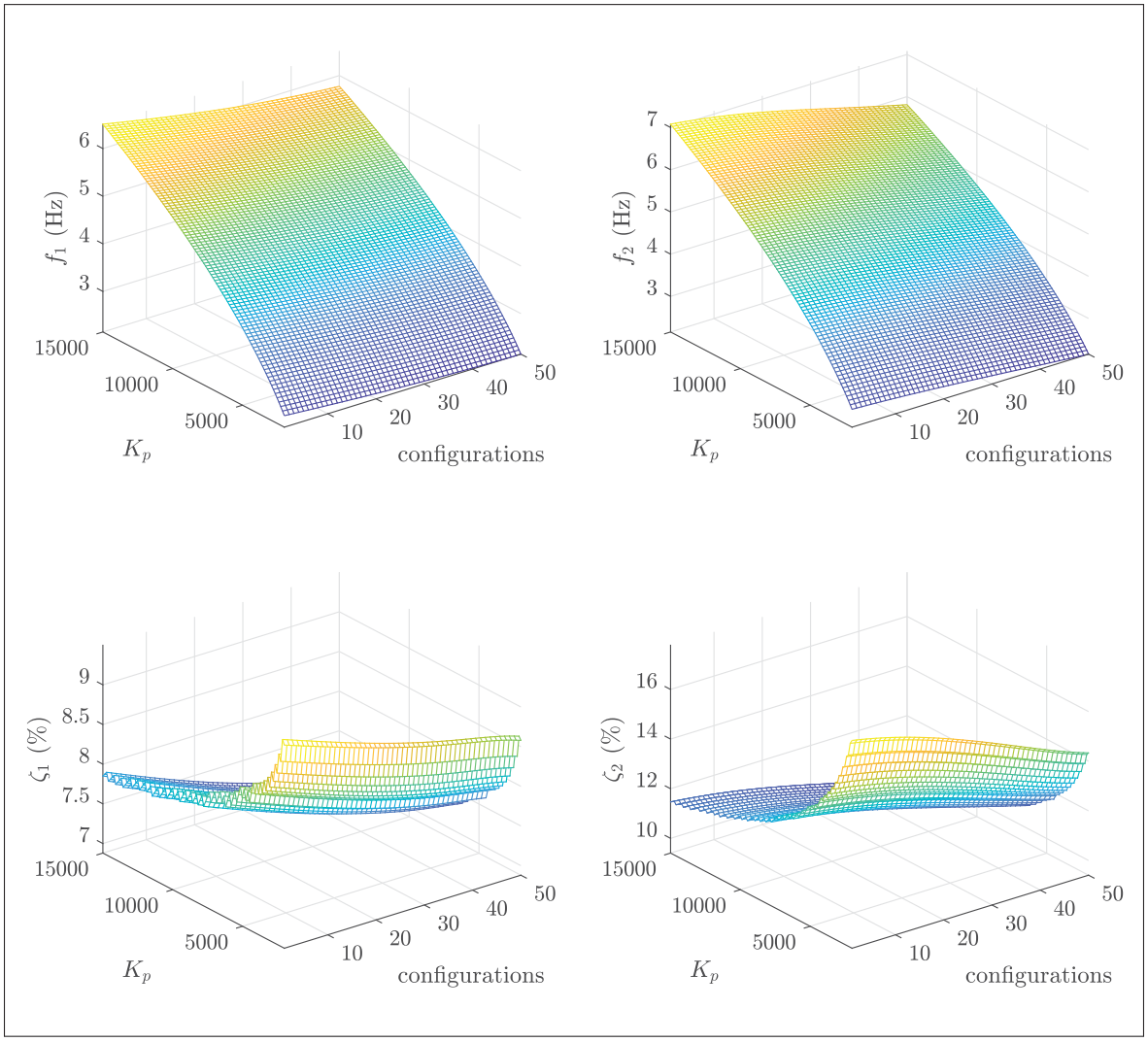

Figure 2.5 Frequencies and damping ratios depending on controller gains

Furthermore, the influence of controller gains on natural frequencies and damping ratios of the first and second modes (case 1) is shown in Fig. 2.5. For numerical simulations, diagonal elements of the matrix $\boldsymbol{K}_{\mathrm{P}}$ are set to be equal and changed linearly from 2000 to 15000, $\boldsymbol{K}_{\mathrm{D}} = 0.425\sqrt{\boldsymbol{K}_{\mathrm{P}}}$, and other parameters are hold constant. For example, at the first configuration, the first frequency rises from 2.39 to 6.51 (Hz), the first damping ratio declines from 9.44 to 7.85 (%). For these modes, when increasing diagonal elements of $\boldsymbol{K}_{\mathrm{P}}$, natural frequencies of the Scompi robot increase, while its damping ratios decrease.

The effect of the control parameters on the first and second mode shapes (case 1) is shown in Fig. 2.6 with two values of controller gains $K_{\mathrm{P},ii} = 12000$ and 15000 ($i = 1, \ldots, 6$). Because eigenvalues obtained from Eq. (2.18) are complex vectors, they are therefore presented in polar coordinates in which one component of an eigenvector is described by a phase and a logarithmic magnitude. Note that in Fig. 2.6, the first six components of eigenvectors corresponding to modal coordinates of six joints are plotted.

2.4.3 Validation

To validate the Scompi's modal parameters obtained using the analytical model presented in Section 2.4.2, the OMA technique based on the VAR method in the time domain is applied (Vu, Thomas, Lafleur & Marcouiller, 2013; Vu *et al.*, 2016). The most advantage of the OMA technique is that modal parameters can be extracted from output-only response, such as position, velocity and acceleration signals. The input responses (excitation forces/moments) do not need to be measured because they are considered as white noise processes (Brincker & Ventura, 2015).

In order to produce the output responses, an impulse force is applied at the end-effector along y_0-axis with the amplitude 50 (N) in the time interval $t \in [0.2,\ 0.2 + 6\Delta t]$ (s) with $\Delta t = 0.001$ (s). The PD controller with gravity compensation described by Eq. (2.2) is used to compute the motor torques for vibration reduction under the external force (see Fig. A I-1 and Fig. A I-2 in

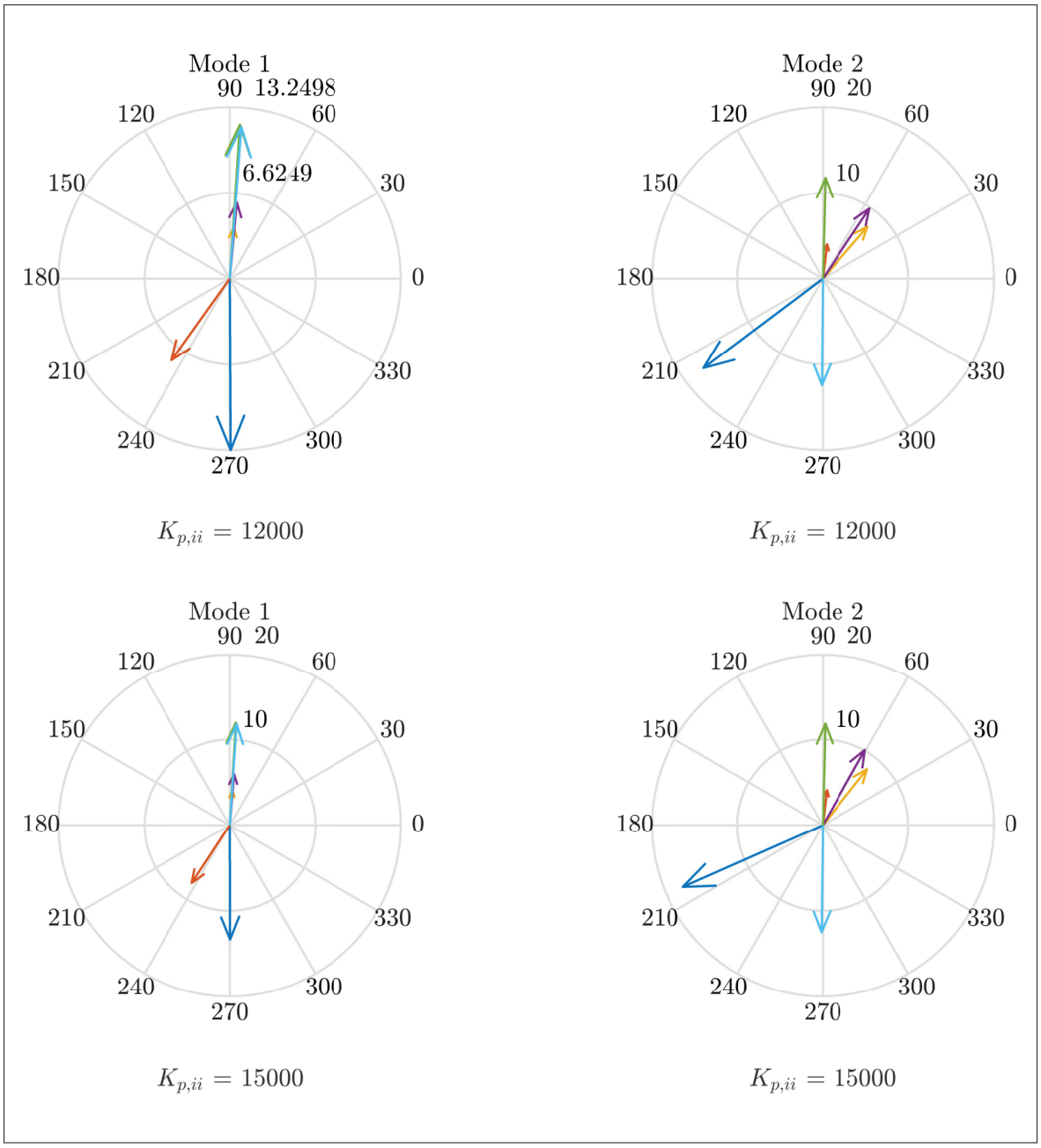

Figure 2.6 Complex mode shapes depending on controller gains

Appendix I.2). Therefore, the output responses can be obtained by solving the direct dynamics problem described in Eq. (2.1) using numerical integration methods, e.g., ode-solvers in Matlab.

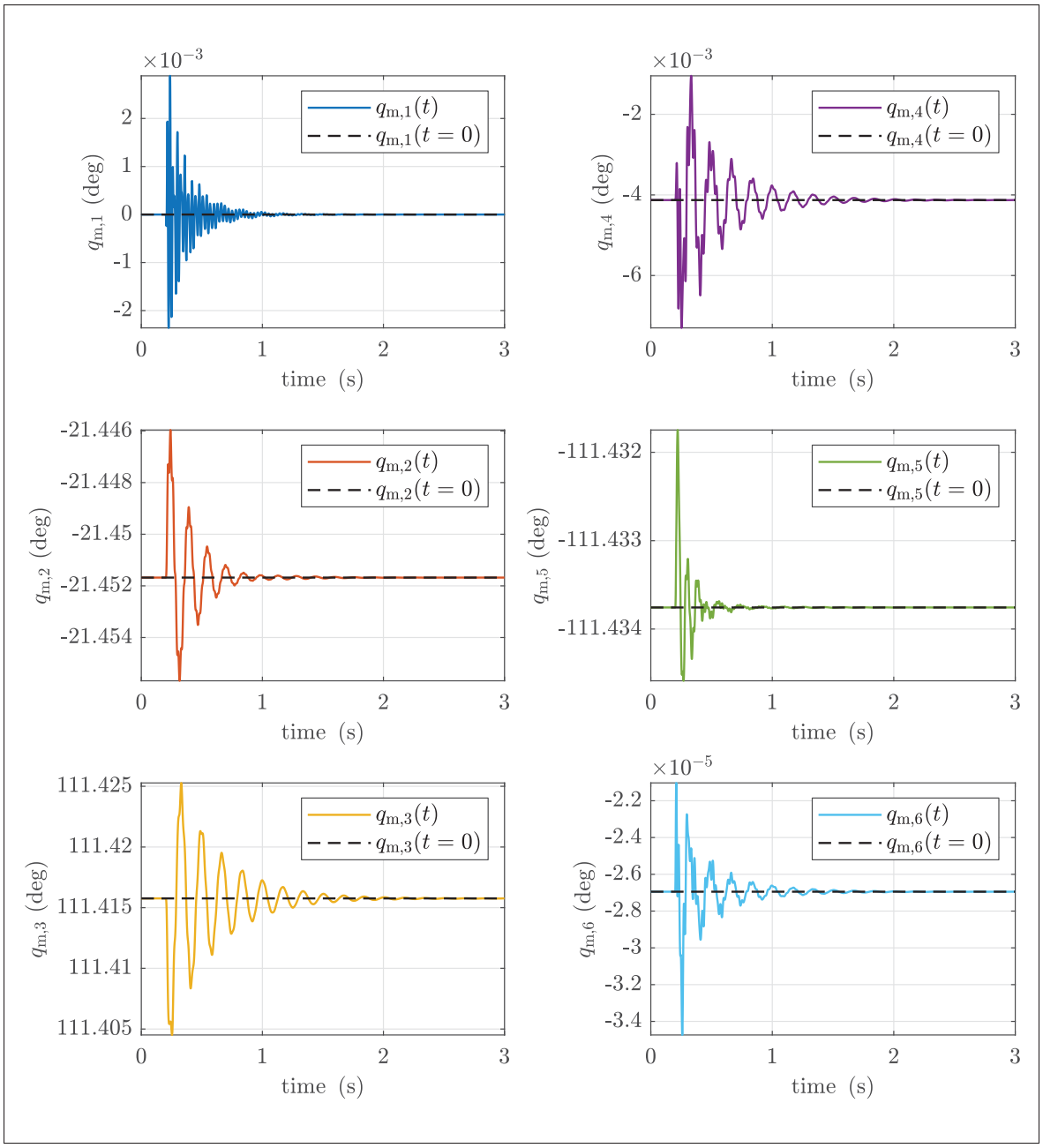

Figure 2.7 Motor positions (case 2)

For the direct dynamics, an initial condition of joint coordinates at $t = 0$ (s) can be selected from 50 configurations presented in Fig. 2.2. For example, if the 8-th configuration is chosen as an

initial condition, then we have:

$$\boldsymbol{q}(0) = [0.0,\ -0.374,\ 1.9448,\ 0.0,\ -1.9448,\ 0.0]^{\mathsf{T}} \tag{2.21}$$

while the initial condition of motor coordinates is computed from Eq. (2.3) with zero external forces/moments:

$$\boldsymbol{q}_{\mathrm{m}}(0) = [0.0,\ -0.374,\ 1.9446,\ -0.0001,\ -1.9449,\ 0.0]^{\mathsf{T}} \tag{2.22}$$

At $t = 0$ (s), we assume that $\dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_{\mathrm{m}}(0) = \mathbf{0}$.

Using the function ode45 in Matlab with the fixed step integration $\Delta t = 0.001$ (s), motor positions (angles) with respect to time are shown in Fig. 2.7. It can be seen that motor positions are reduced exponentially to the desired values for a short time using the feedforward+PD controller. From motor positions (output responses), the Scompi robot's modal parameters can be identified using the VAR method. Note that motor positions are used to identify modal parameters because they can be measured very accurately using encoders integrated into all motor drivers.

To estimate modal parameters, a very high model order range, the order from 2 to 100, is investigated. For each model order, matrices of autoregressive parameters (ARs) are firstly computed using pseudo-inverse method based on QR or SVD techniques. Next, a discrete state-space matrix for eigenvalue problem is derived from ARs (Vu, Thomas, Lakis & Marcouiller, 2011). Therefore, modal parameters including undamped natural frequencies, damping ratios and complex mode shapes can be extracted from eigenvalues and eigenvectors of the discrete state-space matrix. For example, stabilization diagrams describing all frequencies and damping ratios versus the model orders are illustrated in Fig. 2.8 (case 1) and Fig. 2.9 (case 2). In addition, the information about complex mode shapes is used to eliminate unstable frequencies/damping ratios using single-mode validation criteria proposed in Reynders *et al.* (2012) such as the relative eigenfrequency and damping ratio differences, modal assurance criterion (MAC), as well as the order modal assurance (OMAC) (Vu *et al.*, 2011).

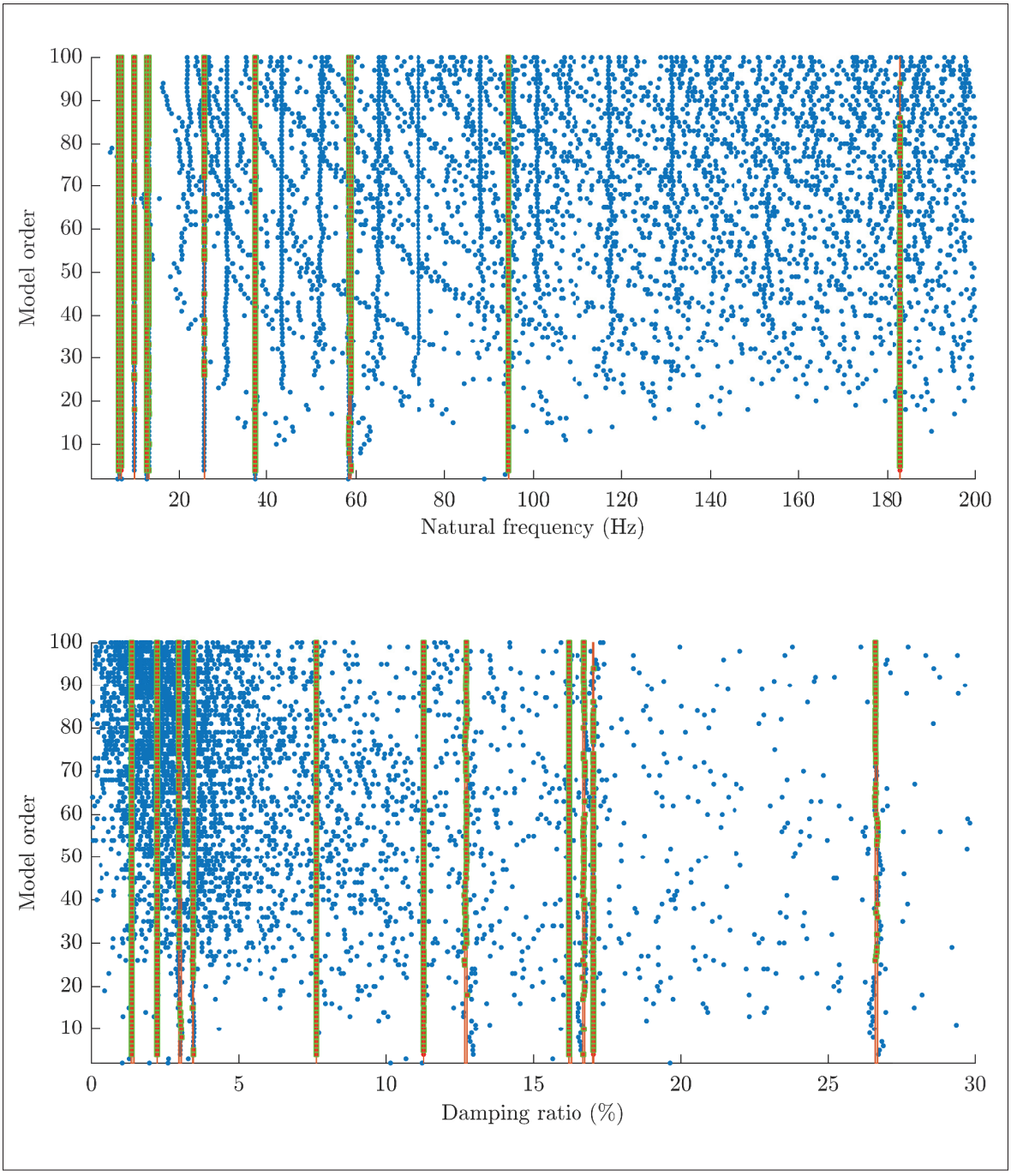

Figure 2.8 Stable natural frequencies and damping ratios (green color) are identified from the motor positions (case 1)

To identify physical modes, i.e., modes appearing in complex conjugate pairs, the grid algorithm implemented in the NAFID-tool is applied (Do, Vu & Liu, 2019). The main idea of this

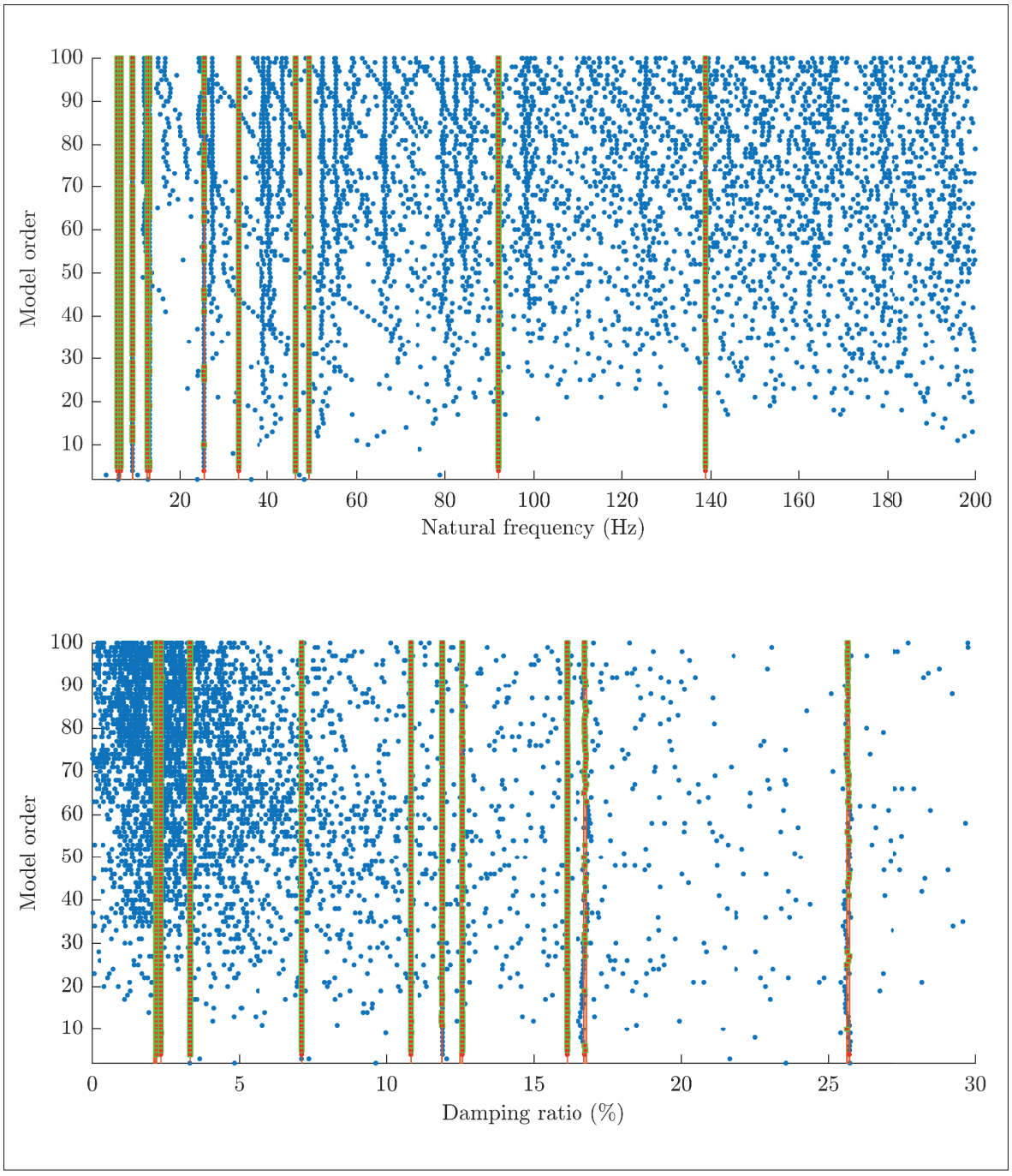

Figure 2.9 Stable natural frequencies and damping ratios (green color) are identified from the motor positions (case 2)

algorithm is to automatically search frequencies/damping ratios with small deviations in the

given ranges (frequency/damping resolutions) along the model-order axis. If the number of frequencies/damping values found is equal or greater than the number of repeating frequencies defined by the user, then frequencies/damping ratios in these ranges are considered to be stable. In NAFID-tool, the default values of frequency and damping resolutions are 0.1 (Hz) and 0.1 (%), respectively. For example, the red points marked by green squares in Fig. 2.9 represent stable vibration modes because their undamped frequencies and damping ratios are nearly constant along the model-order axis. The blue points are unstable frequencies/damping ratios due to computational error or noise.

Table 2.5 Frequencies and damping ratios corresponding to the 8-th configuration obtained using AMA and OMA methods

	AMA-Case 1		OMA-Case 1		AMA-Case 2		OMA-Case 2	
mode	f (Hz)	ζ (%)	f (Hz)	ζ (%)	f (Hz)	ζ (%)	f (Hz)	ζ (%)
1	6.3692	7.6686	6.4191	7.6685	5.9634	7.1538	6.0133	7.1538
2	6.9642	11.2981	6.9191	11.3001	6.6522	10.8175	6.6133	10.8185
3	9.8842	12.7399	9.9191	12.7214	9.2928	11.9109	9.3133	11.8997
4	12.7170	16.1988	12.7170	16.2361	12.6414	16.1256	12.6133	16.1264
5	13.1418	16.7181	13.1191	16.7334	13.1418	16.7181	13.1133	16.7640
6	25.5949	26.5926	25.6191	26.6241	25.3752	25.6469	25.4133	25.6637
7	37.0199	1.3938	37.0197	1.4237	33.1574	2.3513	33.1133	2.3514
8	58.3984	2.9817	58.4191	3.0212	46.3479	2.1881	46.3133	2.1703
9	58.8631	3.4643	58.8191	3.4564	49.3412	3.3381	49.3133	3.3375
10	94.3825	2.2483	94.4191	2.2508	92.0281	2.2052	92.0275	2.1623
11	183.2303	17.0471	183.2191	17.0463	138.7911	12.5831	138.8000	12.5469
12	850.8082	80.5774	–	–	850.5782	80.5557	–	–

The natural frequencies and damping ratios obtained using the analytical and VAR methods are summarized in Table 2.5. There are eleven stable frequencies/damping ratios identified using the NAFID-tool in which the computation time is approximately 120 (s). It can be seen that the results obtained show an excellent matching. However, the 12-th mode is very difficult to identify, even the fixed time step Δt is set to 0.0005 (s) to detect the highest frequency (the Nyquist frequency) up to 1000 (Hz). It can explain by the fact that the last mode has a very high damping ratio, > 80 (%) for both cases.

Furthermore, acceleration signals of the end-effectors are also used to estimate modal parameters (see Fig. A I-2 in Appendix I.2). However, there are eight stable modes found using NAFID-tool (see Fig. A I-3). It can be inferred that the motor position signals may contain more 'useful

information' for the modal parameter estimation than that obtained from the end-effector's acceleration signals. This exploration may be considered a promising aspect of future research.

Overall, numerical simulations prove that the proposed formulation for the modal analysis can be used to compute natural frequencies and damping ratios of flexible-joint robots in the workspace.

2.5 Conclusions

Analytical formulations for the dynamic simulation and computing modal parameters of flexible-joint robots in their workspace are presented in this paper. First, dynamic equations of a general robot with prismatic and revolute joints are established based on the Lagrangian formulation, and then linearized equations for the modal analysis are derived using the Taylor series expansion. Using the proposed formulations, natural frequencies and damping ratios depend on the robot's configuration, the gravity and external forces, and the control parameters. Modal parameters of the lightweight Scompi robot with six flexible joints are computed for various configurations. The results obtained from the analytical method are validated using the operational modal analysis technique based on the vector autoregression model in which the output responses for the identification problem are motor position signals instead of acceleration signals as traditional approaches. The proposed formulations can be used to identify stiffness and damping parameters of flexible joints and to optimize control parameters for vibration suppression. Future works include the effect of cutting forces and other control strategies on computing modal parameters.

CHAPTER 3

IDENTIFICATION OF STIFFNESS AND DAMPING PARAMETERS FOR FLEXIBLE JOINT MANIPULATORS

Thanh Trung Do[1] , Viet Hung Vu[1] , Zhaoheng Liu[1]

[1] Department of Mechanical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted for publication in *Acta Mechanica*, November 2021.

Abstract

Knowledge of the stiffness and damping parameters is critical to understanding the dynamic behavior of robotic manipulators, especially in robotic machining applications. This paper presents an optimization procedure to identify stiffness and damping parameters of joints and motors for manipulators with flexible joints. Based on the linearization model obtained from the dynamic equations of the robot with the PD controller, the measured natural frequencies and damping ratios on different robot configurations, the objective function is derived from the characteristic equation of the eigenvalue problem. Therefore, stiffness and damping parameters are found by solving the optimization problem with bound constraints. Numerical simulations for a robot with two flexible joints are performed to illustrate the applicability of the proposed method. The influence of damping ratio deviation on identified parameters is investigated for several cases.

3.1 Introduction

Dynamic modeling of robotic manipulators for model-based simulation and control requires prior knowledge of physical parameters such as masses, moments of inertia, centers of masses, and spring coefficients and parameters related to friction Chung *et al.* (2016).

For a robot with rigid links and joints, the identification problem of the mentioned parameters above has been presented in many papers such as Swevers, Verdonck & De Schutter (2007); Wu, Wang & You (2010); Hollerbach, Khalil & Gautier (2016) using computer-aided design software (CAD) or through experimental measurements. For the latter, the robot dynamic equations have to be expressed as the product of a regressor matrix (depending on joint positions, speeds and accelerations) and a vector of minimal/base dynamic parameters (grouped inertial parameters). From measured or estimated joint positions, speeds and accelerations, as well as the motor torques, these parameters will be found from the overdetermined linear equations using the pseudo inverse matrix method Swevers, Ganseman, De Schutter & Van Brussel (1996).

For a robot with rigid links and flexible joints, dynamic equations of the robot are quite complicated because the joint and motor coordinates are independent Spong (1987); De Luca & Book (2016). There are several works to consider the parameter identification problem of these robots.

In Zollo, Lopez, Spedaliere, Aracil & Guglielmelli (2015), a general method to identify all dynamic parameters of robots with elastic joints was presented. This method was developed based on the dynamic identification methods for robots with rigid links and joints. The main drawback of this method is that it requires many sensors to measure the motion variables (positions, speeds and accelerations of joints and motors). Note that only motor angle positions are available to measure using encoders for industrial robots. In Miranda-Colorado & Moreno-Valenzuela (2018), a method based on the filtered dynamic model was proposed. However, this method is also required to measure the positions and speeds of joints and motors.

In Ohr *et al.* (2006), the frequency response functions (FRFs) established based on the motor torques, and motor speeds were used to identify flexibility parameters of the 6-axis industrial manipulator with flexible joints. Each joint in the first three joints of the robot were modeled by three degrees of freedom with three-dimensional spring-damper pairs and the other joints were modeled by one degree of freedom with one spring-damper pair. However, the use of spring/damping parameters in the model-based control strategy is not a simple task. In Neubauer, Gattringer, Müller, Steinhauser & Höbarth (2015), authors applied the genetic and gradient-based

minimization algorithms to identify stiffness and damping parameters of the Stäubli TXL robot. The objective function was derived from the measured frequency response matrix, and the motor transfer matrix of the linearized system for the first three joints in which the linearized system used was parameter-independent of the PD controller. In addition, to reduce the effect of the controller on measured input and output response when conducting experimental modal analysis, the authors proposed the use of small feedback gains of the controller. However, this can make the robot unstable in some configurations because of the influence of gravity. A similar approach based on FRFs for identification of stiffness and damping parameters of industrial robots has recently been presented in Huynh, Assadi, Dambly, Rivière-Lorphèvre & Verlinden (2021). By minimizing the error between the measured and simulated FRFs in the sensor frame, a direct method based on the genetic algorithm followed by a deterministic procedure was presented to update stiffness and damping parameters of multibody systems and robotic arms. The stiffness and damping parameters of the first three joints for the KUKA-KR90 R3100 robot were updated successfully using this method.

The main contribution of this work is to present an optimization procedure for determining stiffness and damping parameters of robotic manipulators with flexible joints. Our approach is developed based on the inverse eigenvalue problem for linear systems Chu & Golub (2005); Mottershead & Ram (2006) and the model updating method Friswell & Mottershead (1995). It consists of the following three steps. (1) Modal parameters (natural frequencies and damping ratios) for several robot configurations in working condition (controlled by the PD controller) are identified by using experimental modal analysis (EMA) Ewins (2000); Mejri *et al.* (2013) or operational modal analysis (OMA) technique Zaghbani *et al.* (2013); Vu *et al.* (2011, 2016). (2) The objective function is derived from the characteristic polynomial of the inverse eigenvalue problem, i.e., based on the linearized dynamic model, the measured natural frequencies and damping ratios. (3) The unknown stiffness and damping parameters are therefore determined by solving the optimization problem in Matlab through the nonlinear least-squares algorithms. In this work, we focus on the last two steps while the first step is assumed to be completed.

The main advantages of the proposed method are:

- Natural frequencies and damping ratios in step 1 can be estimated using OMA instead of EMA as most previous researches. This is because our approach does not require measuring the input response (forces/torques). In addition, the motor positions measured using the encoders can be used as the output response for OMA in order to identify modal parameters Do, Vu & Liu (2022).
- The objective function obtained from the characteristic polynomial is more straightforward than other approaches based on the frequency response functions as presented in the literature.
- The linearized dynamic model used to compute modal parameters takes the influence of the PD control gains.

It is important to note that in the mentioned methods above, kinematic and dynamic parameters of rigid links and motors are assumed to be predetermined, such as from manufacturing or CAD data. In addition, the robot configurations are assumed to be measured using internal/external sensors such as laser trackers.

The remainder of this paper is organized as follows. In Section 3.2, the dynamic equations of motion of flexible joint robots with the PD controller are briefly presented using the Lagrangian formulation. In Section 3.3, the direct and inverse eigenvalue problem of flexible joints robots are discussed based on the linearized model. The objective function is derived in Section 3.4 in order to identify stiffness and damping parameters. The application of numerical methods to solve the optimization is also addressed. In Section 3.5, the identification of stiffness and damping parameters for the planar robot with two flexible joints are performed. The effect of noise on optimal solution is investigated. Finally, conclusions and future works are given in Section 3.6.

3.2 Dynamic modeling of flexible joint manipulators with the PD controller

A brief description of the dynamic equations and linearized model of flexible joint robots is presented. For the modeling of a flexible joint robot, we assume that the robot has rigid links and flexible joints.

3.2.1 Dynamic equations

Based on the Lagrangian formulation and the well-known assumption proposed in (Spong, 1987), the equations of motion for a robot consisting of n flexible prismatic and revolute joints and no external forces/moments acting on the end-effector can be expressed as follows (Do *et al.*, 2022):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{W}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) = \boldsymbol{0} \tag{3.1}$$

$$\boldsymbol{B}\ddot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) = \boldsymbol{\tau}_{\mathrm{m}} \tag{3.2}$$

where $\boldsymbol{q} = [q_1,\, q_2, \ldots,\, q_n]^{\mathsf{T}}$ and $\boldsymbol{q}_{\mathrm{m}} = [q_{\mathrm{m},1},\, q_{\mathrm{m},2}, \ldots,\, q_{\mathrm{m},n}]^{\mathsf{T}}$ are vectors of joint coordinates and motor coordinates (angle positions of motors after reduction), respectively.

It can be seen that, Eq. (3.1) describes the dynamics of rigid links where $\boldsymbol{M}$ and $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ represent mass and centrifugal Coriolis matrices, respectively, and $\boldsymbol{g} \in \mathbb{R}^n$ is the gravity vector, while Eq. (3.2) describes the dynamics of motors where $\boldsymbol{B} = \mathrm{diag}(N_1^2 I_{\mathrm{r}_1,z}, N_2^2 I_{\mathrm{r}_2,z}, \ldots, N_n^2 I_{\mathrm{r}_n,z})$ is the diagonal inertia matrix of motors in which $I_{\mathrm{r}_i,z}$ is the inertia moment of motor i about its rotating z-axis and N_i is the gear ratio. The matrix $\boldsymbol{W}$ is defined by: $\boldsymbol{W} = \mathrm{diag}(\frac{1}{r_1}, \frac{1}{r_2}, \ldots, \frac{1}{r_n})$, where r_i describes the characteristic radius of joint driver i. Note that $r_i = 1$ if the joint i is revolute. Therefore, if the robot has flexible revolute joints (De Luca & Tomei, 1996; Ott, 2008; De Luca & Book, 2016), then $\boldsymbol{W}$ is the identity matrix.

The stiffness and damping matrices of joints and motors are defined as follows:

$$\boldsymbol{K} = \mathrm{diag}(k_1, k_2, \ldots, k_n), \tag{3.3}$$

$$\boldsymbol{D} = \mathrm{diag}(d_{\mathrm{f},1}, d_{\mathrm{f},2}, \ldots, d_{\mathrm{f},n}) \tag{3.4}$$

$$\boldsymbol{D}_{\mathrm{m}} = \mathrm{diag}(d_{\mathrm{fm},1}, d_{\mathrm{fm},2}, \ldots, d_{\mathrm{fm},n}) \tag{3.5}$$

where k_i is the stiffness parameter of joint i, while $d_{\mathrm{f},i}$ and $d_{\mathrm{fm},1}$ are viscous coefficients of joint i and motor i, respectively.

Based on the PD control with constant gravity compensation proposed in (Tomei, 1991), the vector of motor torques $\boldsymbol{\tau}_{\mathrm{m}}$ is given by (Do *et al.*, 2022)

$$\boldsymbol{\tau}_{\mathrm{m}} = \boldsymbol{W}^{-1}\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) + \boldsymbol{K}_{\mathrm{P}}(\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} - \boldsymbol{q}_{\mathrm{m}}) - \boldsymbol{K}_{\mathrm{D}}\dot{\boldsymbol{q}}_{\mathrm{m}} \tag{3.6}$$

where $\boldsymbol{K}_{\mathrm{P}}$ and $\boldsymbol{K}_{\mathrm{D}}$ are, respectively, proportional and derivative gain matrices, and $\boldsymbol{q}^{\mathrm{d}}$ and $\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}}$ are the vectors of the desired joint and motor positions, respectively, in which $\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}}$ is determined by:

$$\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} = \boldsymbol{W}\boldsymbol{q}^{\mathrm{d}} + (\boldsymbol{K}\boldsymbol{W})^{-1}\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) \tag{3.7}$$

Substituting Eq. (3.6) into Eq. (3.1) and Eq. (3.2), one leads to the closed-loop control system:

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{W}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) \\ \boldsymbol{B}\ddot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} - \boldsymbol{K}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) - \boldsymbol{W}^{-1}\boldsymbol{g}(\boldsymbol{q}^{\mathrm{d}}) - \boldsymbol{K}_{\mathrm{P}}(\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}} - \boldsymbol{q}_{\mathrm{m}}) + \boldsymbol{K}_{\mathrm{D}}\dot{\boldsymbol{q}}_{\mathrm{m}} \end{bmatrix} = \boldsymbol{0} \tag{3.8}$$

3.2.2 Linearized dynamic model

By using the notation $\boldsymbol{z} \triangleq [\boldsymbol{q}^{\mathsf{T}}, \boldsymbol{q}_{\mathrm{m}}^{\mathsf{T}}]^{\mathsf{T}}$ and the Taylor series expansion (Ginsberg, 1998; Lynch & Vanderploeg, 1995), the linearized model of a flexible joint robot about the desired equilibrium position $\boldsymbol{z}^{\mathrm{d}} = [(\boldsymbol{q}^{\mathrm{d}})^{\mathsf{T}}, (\boldsymbol{q}_{\mathrm{m}}^{\mathrm{d}})^{\mathsf{T}}]^{\mathsf{T}}$ and $\dot{\boldsymbol{z}}^{\mathrm{d}} = \boldsymbol{0}, \ddot{\boldsymbol{z}}^{\mathrm{d}} = \boldsymbol{0}$ is derived from Eq. (3.8) as follows (Do *et al.*, 2022):

$$\overline{\boldsymbol{M}}\Delta\ddot{\boldsymbol{z}} + \overline{\boldsymbol{D}}\Delta\dot{\boldsymbol{z}} + \overline{\boldsymbol{K}}\Delta\boldsymbol{z} = \boldsymbol{0} \tag{3.9}$$

where $\Delta\ddot{\boldsymbol{z}} = \ddot{\boldsymbol{z}} - \ddot{\boldsymbol{z}}^{\mathrm{d}}$, $\Delta\dot{\boldsymbol{z}} = \dot{\boldsymbol{z}} - \dot{\boldsymbol{z}}^{\mathrm{d}}$ and $\Delta\boldsymbol{z} = \boldsymbol{z} - \boldsymbol{z}^{\mathrm{d}}$ represent the perturbation vectors, and the three linearized matrices $\overline{\boldsymbol{M}}$, $\overline{\boldsymbol{D}}$, and $\overline{\boldsymbol{K}} \in \mathbb{R}^{2n\times 2n}$ in Eq. (3.9) are defined as:

$$\overline{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}^{\mathrm{d}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}, \tag{3.10}$$

$$\overline{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{m}} + \boldsymbol{K}_{\mathrm{D}} \end{bmatrix}, \tag{3.11}$$

$$\overline{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_{\mathrm{G}}(\boldsymbol{q}^{\mathrm{d}}) + \boldsymbol{K}\boldsymbol{W}^2 & -\boldsymbol{K}\boldsymbol{W} \\ -\boldsymbol{K}\boldsymbol{W} & \boldsymbol{K} + \boldsymbol{K}_{\mathrm{P}} \end{bmatrix} \tag{3.12}$$

and $\boldsymbol{K}_{\mathrm{G}} \in \mathbb{R}^{n\times n}$ is the gradient of $\boldsymbol{g}$ with respect to $\boldsymbol{q}$:

$$\boldsymbol{K}_{\mathrm{G}}(\boldsymbol{q}) = \frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{3.13}$$

The following properties can be observed from Eqs. (3.10−3.13):

- The linearized mass matrix $\overline{\boldsymbol{M}}$ is symmetric positive definite and it depends on the desired configuration $\boldsymbol{q}^{\mathrm{d}}$. i.e., when the robot changes the configuration, elements of this matrix will be varied.
- The linearized damping matrix $\overline{\boldsymbol{D}}$ is symmetric and depends on damping parameters of joints/motors, as well as derivative gains of the controller.
- The linearized stiffness matrix $\overline{\boldsymbol{K}}$ is symmetric and depends on not only $\boldsymbol{q}^{\mathrm{d}}$, but also joint stiffness parameters and proportional gains of the controller.

One of the most advantages of the linearized model described by Eq. (3.9) is that it allows us to compute modal parameters of flexible joint robots for any desired configuration in operating conditions. When gain parameters of the PD controller are adjusted while other parameters remain constant, the natural frequencies and damping ratios of the robot will change. In other words, the PD controller plays the role of a virtual spring-damper system in which the gain parameters are similar to spring and damping coefficients.

3.3 Direct and inverse eigenvalue problems

Based on the linearized model derived from the previous section, two problems are introduced: the direct eigenvalue and inverse eigenvalue problems of flexible joint robots. We assume that all constant kinematic/dynamic parameters of links and motors (masses, positions of center of masses, inertia tensors, gear ratios, and gain parameters) are predetermined.

3.3.1 The direct eigenvalue problem of flexible-joint robots

For the direct eigenvalue problem, given the stiffness and damping parameters as well as the desired configuration $\boldsymbol{q}^{\mathrm{d}}$, the eigenvalues and eigenvectors (mode shapes) can be computed from the linearized matrices defined in Eq. (3.9).

Because the damping matrix $\overline{\boldsymbol{D}}$ described by Eq. (3.11) is non-proportional, in which $\overline{\boldsymbol{D}} \neq \alpha\overline{\boldsymbol{M}} + \beta\overline{\boldsymbol{K}}$ where α and β are constants of proportionality, modal parameters of flexible-joint robots can be obtained using the state-space formulation presented in (Do *et al.*, 2022). In this work, the quadratic eigenvalue formulation for the general eigenvalue problem (Inman, 2017) is used for computing eigenvalues, as well as for the identification parameter problem discussed in Section 3.3.2 . From Eq. (3.9), the eigenvalue problem can be formulated as follows (Inman, 2017):

$$\left(\overline{\boldsymbol{M}}\lambda_k^2 + \overline{\boldsymbol{D}}\lambda_k + \overline{\boldsymbol{K}}\right)\boldsymbol{u}_k = \boldsymbol{0} \tag{3.14}$$

where λ_k and $\boldsymbol{u}_k$ are an eigenvalue and an eigenvector, respectively. Since the mass matrix $\overline{\boldsymbol{M}}$ is symmetric positive definite, the existence of a matrix square root $\overline{\boldsymbol{S}}$ fulfills (Inman, 2017):

$$\overline{\boldsymbol{M}} = \overline{\boldsymbol{S}}\,\overline{\boldsymbol{S}} \tag{3.15}$$

where $\overline{\boldsymbol{S}}$ is also a positive matrix. By pre-multiplying Eq. (3.14) with an inversion matrix $(\overline{\boldsymbol{S}})^{-1}$, and replacing $\boldsymbol{u}_k$ by $(\overline{\boldsymbol{S}})^{-1}\overline{\boldsymbol{S}}\,\boldsymbol{u}_k$, after some manipulations, Eq. (3.14) can be rearranged as

follows:

$$\underbrace{\left(\boldsymbol{I}\lambda_k^2 + (\overline{\boldsymbol{S}})^{-1}\overline{\boldsymbol{D}}(\overline{\boldsymbol{S}})^{-1}\lambda_k + (\overline{\boldsymbol{S}})^{-1}\overline{\boldsymbol{K}}(\overline{\boldsymbol{S}})^{-1}\right)}_{\overline{\boldsymbol{A}}_k}\underbrace{\overline{\boldsymbol{S}}\,\boldsymbol{u}_k}_{\boldsymbol{v}_k} = \boldsymbol{0} \tag{3.16}$$

where $\overline{\boldsymbol{A}}_k$ is the polynomial matrix and $\boldsymbol{v}_k$ is an intermediate eigenvector. For a nontrivial solution $\boldsymbol{v}_k$, an eigenvalue λ_k must be a root of the characteristic polynomial of order $4n$:

$$|\overline{\boldsymbol{A}}_k| \triangleq \det\left(\boldsymbol{I}\lambda_k^2 + (\overline{\boldsymbol{S}})^{-1}\overline{\boldsymbol{D}}(\overline{\boldsymbol{S}})^{-1}\lambda_k + (\overline{\boldsymbol{S}})^{-1}\overline{\boldsymbol{K}}(\overline{\boldsymbol{S}})^{-1}\right) = 0 \tag{3.17}$$

where $\boldsymbol{I} \in \mathbb{R}^{2n\times 2n}$ denotes the identity matrix. By solving Eq. (3.17) using numerical methods, we will have $4n$ solutions in real and/or complex numbers (usually complex conjugate pairs). Once a solution λ_k is obtained from Eq. (3.17), a corresponding eigenvalue $\boldsymbol{v}_k$ can be determined from Eq. (3.16), and therefore, $\boldsymbol{u}_k = (\overline{\boldsymbol{S}})^{-1}\boldsymbol{v}_k$.

If a complex eigenvalue λ_k has a negative real part (the under-damped case), then a natural frequency ω_k (> 0) and a damping ratio ζ_k (< 1) are computed as follows:

$$\omega_k = \sqrt{\mathrm{Re}(\lambda_k)^2 + \mathrm{Im}(\lambda_k)^2} \quad \text{(rad/s)}, \tag{3.18}$$

$$\zeta_k = -\frac{\mathrm{Re}(\lambda_k)}{\omega_k} \tag{3.19}$$

where $\mathrm{Re}(\lambda_k)$ and $\mathrm{Im}(\lambda_k)$ are the real and imaginary parts of λ_k. In this case, $\boldsymbol{u}_k$ is a complex eigenvector.

3.3.2 The inverse eigenvalue problem of flexible-joint robots

Generally, the purpose of the inverse eigenvalue problem is to determine the mass, damping and stiffness matrices based on the completed or partitioned natural frequencies, damping ratios and mode shapes measured using experiments (Chu & Golub, 2005).

In this work, the inverse eigenvalue problem of flexible joint robots is to find the approximate values of the stiffness and damping parameters of joints and motors. To solve this problem, we

assume that several natural frequencies and damping ratios at different robot configurations are measured, e.g., using the OMA technique.

Note that in practice, while natural frequencies can be identified with great accuracy, damping ratios are more difficult to determine than frequencies because they are sensitive to noise. For example, if the stabilization diagram is used to identify stable modes (Cara, Juan, Alarcon, Reynders & De Roeck, 2013), the criteria used to identify stable natural frequencies, damping ratios and mode shapes are: 2% for the frequency, 5% for the damping ratio, and 5% for the mode shape. For each specific case, the criterion for the identification of the damping ratios may be higher.

From N measured natural frequencies $\hat{\omega}_k$ and damping ratios $\hat{\zeta}_k$, the corresponding eigenvalues $\hat{\lambda}_k$ for $k = 1, \ldots, N$ can be expressed as:

$$\hat{\lambda}_k = -\hat{\omega}_k \hat{\zeta}_k + j\hat{\omega}_k \sqrt{1 - \hat{\zeta}_k^2} \tag{3.20}$$

where $j^2 = -1$. By substituting $\lambda_k = \hat{\lambda}_k$ into the left-hand side of Eq. (3.17), one obtains

$$|\hat{\boldsymbol{A}}_k| = \det\left(\boldsymbol{I}\hat{\lambda}_k^2 + (\overline{\boldsymbol{S}})^{-1}\hat{\boldsymbol{D}}(\overline{\boldsymbol{S}})^{-1}\hat{\lambda}_k + (\overline{\boldsymbol{S}})^{-1}\hat{\boldsymbol{K}}(\overline{\boldsymbol{S}})^{-1}\right) \tag{3.21}$$

where $\hat{\boldsymbol{A}}_k$ represents the polynomial matrix of the inverse eigenvalue problem. Note that the diagonal elements of matrices $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{K}}$ depend on unknown stiffness and damping parameters, while the matrix $\overline{\boldsymbol{S}}$ is a constant because it depends on the given robot configuration.

Theoretically, if $\hat{\lambda}_k$ is a true solution of the characteristic polynomial $|\hat{\boldsymbol{A}}_k|$, then the real and imaginary parts of $|\hat{\boldsymbol{A}}_k|$ must be equal to zero:

$$\mathrm{Re}(|\hat{\boldsymbol{A}}_k|) = 0, \quad \text{and} \quad \mathrm{Im}(|\hat{\boldsymbol{A}}_k|) = 0. \tag{3.22}$$

For $k = 1, \ldots, N$, we can establish an over-determined nonlinear system from Eq. (3.22) in order to determine the unknown parameters.

In practice, the equality conditions in Eq. (3.22) are impossible to achieve because $\hat{\lambda}_k$ is affected by noise. Therefore, we propose that the condition Eq. (3.22) is rewritten as follows:

$$\left(\mathrm{Re}(|\hat{\boldsymbol{A}}_k|)\right)^2 + \left(\mathrm{Im}(|\hat{\boldsymbol{A}}_k|)\right)^2 \rightarrow 0. \tag{3.23}$$

3.4 Identification of stiffness and damping parameters

This section presents an optimization procedure based on the inverse eigenvalue problem to identify the stiffness and damping parameters of joints and motors.

3.4.1 Selection of modal parameters

We assume that n_c different robot configurations are chosen for conducting modal testing in order to estimate modal parameters. Note that singular configurations (Siciliano *et al.*, 2009; Angeles, 2014) (the geometric Jacobian matrix depending on the joint coordinates is singular or not full rank) must be avoided to ensure that the robot is stable when conducting experiments. For the i-th configuration, we assume that n_{c_i} stable modes of vibrations are identified using modal analysis tools. However, only n_{s_i} natural frequencies $\hat{\omega}_{i,k}$ and n_{s_i} damping ratios $\hat{\zeta}_{i,k}$ for $k = 1, \ldots, n_{\mathrm{s}_i}$ and $n_{\mathrm{s}_i} \leq n_{\mathrm{c}_i}$ are used for the optimization problem. These frequencies should be selected in the lower frequency range, as they contain the basic information of the system and seem to be less affected by noise than high frequencies. For example, the fundamental frequency of industrial robotics is identified from 10 to 14 (Hz) (Iglesias, Sebastián & Ares, 2015). In addition, the higher frequencies related to the resonances of robot links (Bisu, Cherif, Gérard & KNevez, 2012) are not to be selected.

Consequently, complex eigenvalues are expressed as:

$$\hat{\lambda}_{i,k} = -\hat{\omega}_{i,k}\hat{\zeta}_{i,k} + j\hat{\omega}_{i,k}\sqrt{1-\hat{\zeta}_{i,k}^2} \tag{3.24}$$

and the total number of eigenvalues for n_c configurations is given by:

$$N = \sum_{i=1}^{n_c} n_{s_i} \tag{3.25}$$

3.4.2 Optimization problem

In order to find the vectors of the stiffness and damping parameters defined by $\boldsymbol{k} \in \mathbb{R}^n$ and $\boldsymbol{d} \in \mathbb{R}^{2n}$:

$$\boldsymbol{k} = [k_1,\ k_2, \ldots, k_n]^\mathsf{T} \tag{3.26}$$

$$\boldsymbol{d} = [d_{\mathrm{f},1}, d_{\mathrm{f},2}, \ldots, d_{\mathrm{f},n}, d_{\mathrm{fm},1}, d_{\mathrm{fm},2}, \ldots, d_{\mathrm{fm},n}]^\mathsf{T} \tag{3.27}$$

the following objective function, $F(\boldsymbol{k}, \boldsymbol{d})$ defined based on Eq. (3.23) is minimized:

$$F(\boldsymbol{k}, \boldsymbol{d}) = \sum_{i=1}^{n_c} \sum_{k=1}^{n_{s_i}} \left(\mathrm{Re}(|\hat{\boldsymbol{A}}_{i,k}|)\right)^2 + \left(\mathrm{Im}(|\hat{\boldsymbol{A}}_{i,k}|)\right)^2 \tag{3.28}$$

subject to:

$$\boldsymbol{0} < \boldsymbol{k} \leq \boldsymbol{k}_{\max} \tag{3.29}$$

$$\boldsymbol{0} < \boldsymbol{d} \leq \boldsymbol{d}_{\max} \tag{3.30}$$

In Eq. (3.28), the terms $\mathrm{Re}(|\hat{\boldsymbol{A}}_{i,k}|)$ and $\mathrm{Im}(|\hat{\boldsymbol{A}}_{i,k}|)$ represent the real and imaginary parts of $|\hat{\boldsymbol{A}}_{i,k}|$, respectively. Here, the matrix $\hat{\boldsymbol{A}}_{i,k}$ is established based on Eq. (3.21), but with the scale factor of $1/\hat{\omega}_{i,k}$. Therefore, $|\hat{\boldsymbol{A}}_{i,k}|$ is defined as follows:

$$|\hat{\boldsymbol{A}}_{i,k}| = \det\left(\frac{\boldsymbol{I}\hat{\lambda}_{i,k}^2 \ +\ (\overline{\boldsymbol{S}}_i)^{-1}\hat{\boldsymbol{D}}_i(\overline{\boldsymbol{S}}_i)^{-1}\hat{\lambda}_{i,k} \ +\ (\overline{\boldsymbol{S}}_i)^{-1}\hat{\boldsymbol{K}}_i(\overline{\boldsymbol{S}}_i)^{-1}}{\hat{\omega}_{i,k}}\right) \tag{3.31}$$

Proper selection of the objective function and the scale factor is crucial to finding the optimal solution. Numerical simulations presented in Section 3.5 will prove the efficiency of the proposed objective function.

The upper bounds of the stiffness and damping parameters in Eq. (3.29) and Eq. (3.30) can generally be set arbitrarily. However, the upper bound of $\boldsymbol{k}$ can be obtained from prior knowledge, e.g., from the manufacturer or using measurements based on the kinostatic approach (Dumas *et al.*, 2011).

In order to obtain the best approximate solution the number of distinct eigenvalues N used in Eq. (3.28) should be greater than the total number of unknowns:

$$\dim(\boldsymbol{k}) + \dim(\boldsymbol{d}) < N \tag{3.32}$$

3.4.3 Optimization method

The nonlinear optimization problem with bound constraints presented above can be solved using the lsqnonlin function in Matlab. Because the objective function is derived from the determinant of complex matrices, its gradient (first-order partial derivatives of the objective function with respect to parameter vectors $\boldsymbol{k}$ and $\boldsymbol{d}$) is difficult to compute analytically. Therefore, the numerical differentiation method is applied to compute the approximate gradient.

Furthermore, to increase the opportunity of finding a global solution, we recommend using the MultiStart function implemented in Matlab Global Optimization Toolbox (MathWorks, 2020b). This function allows us to support many initial values randomly assigned or predetermined by the user. Computationally, MultiStart will attempt to search many local solutions based on a matrix of initial values in which each row of this matrix represents a vector of initial values. Based on the local solutions, the best/optimal solution $(\boldsymbol{k}^\star, \boldsymbol{d}^\star)$ is determined if the objective function $F(\boldsymbol{k}^\star, \boldsymbol{d}^\star)$ is minimal. Note that when multiple initial values are used to find an optimal

solution, we can apply parallel computation for a multi-core computer to reduce the computation time.

3.5 Numerical simulations

In this section, six stiffness and damping parameters of the planar flexible joint robots are identified using the above procedure. The robot's physical parameters and modal parameters are assumed to be predetermined.

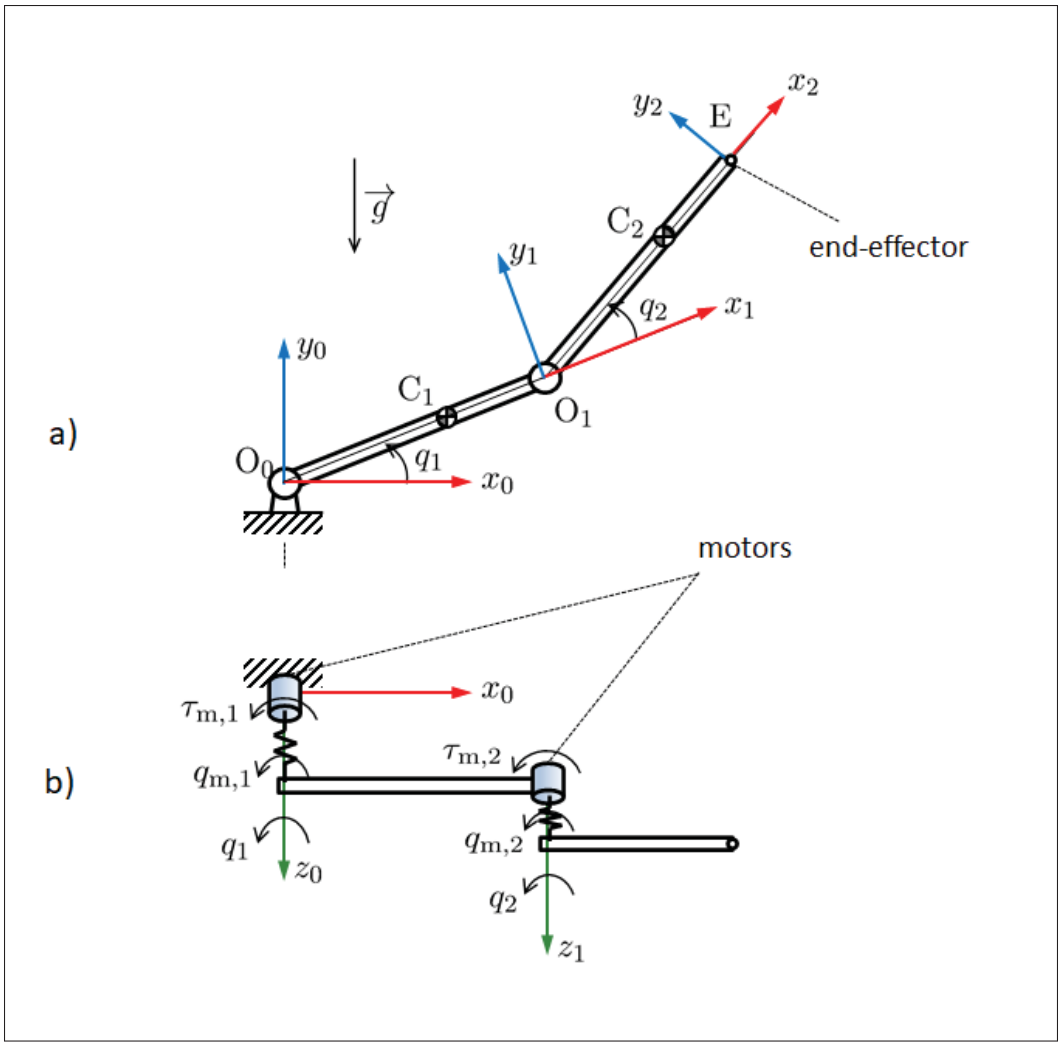

Figure 3.1 Model of the flexible joint robot: a) front view and b) top view

3.5.1 Robot description

The planar robot shown in Fig. 3.1 includes two rigid links and two flexible joints. Four generalized coordinates are used to establish dynamic equations of motion: the vector of joint coordinates $\boldsymbol{q} = [q_1, q_2]^\mathsf{T}$, and the motor coordinates $\boldsymbol{q}_\mathrm{m} = [q_{\mathrm{m},1}, q_{\mathrm{m},2}]^\mathsf{T}$.

Based on the Lagrangian formulation, the dynamic equations and a linearized model of the robot can be obtained by using computer algebra systems, such as the symbolic Maple software. For example, the mass matrix and gravity vector of the links are computed as follows:

$$\boldsymbol{M} = \begin{bmatrix} I_1 + I_2 + m_1 x_{\mathrm{C}_1}^2 + m_2(a_1^2 + x_{\mathrm{C}_2}^2) + 2m_2 a_1 x_{\mathrm{C}_2} \cos q_2 & m_2 a_1 x_{\mathrm{C}_2} \cos q_2 \\ m_2 a_1 x_{\mathrm{C}_2} \cos q_2 & I_2 + m_2 x_{\mathrm{C}_2}^2 \end{bmatrix}, \tag{3.33}$$

$$\boldsymbol{g} = \begin{bmatrix} g(m_1 x_{\mathrm{C}_1} + m_2 a_1) \cos q_1 + g m_2 x_{\mathrm{C}_2} \cos q_{12} \\ g m_2 x_{\mathrm{C}_2} \cos q_{12} \end{bmatrix} \tag{3.34}$$

To establish the linearized dynamic model, the matrix $\boldsymbol{K}_\mathrm{G}$ in Eq. (3.12) is derived from $\boldsymbol{g}$ as follows:

$$\boldsymbol{K}_\mathrm{G} = -g \begin{bmatrix} (m_1 x_{\mathrm{C}_1} + m_2 a_1) \sin q_1 + m_2 x_{\mathrm{C}_2} \sin q_{12} & m_2 x_{\mathrm{C}_2} \sin q_{12} \\ m_2 x_{\mathrm{C}_2} \sin q_{12} & m_2 x_{\mathrm{C}_2} \sin q_{12} \end{bmatrix} \tag{3.35}$$

where $q_{12} = q_1 + q_2$ and $g = 9.81\ (\mathrm{m \cdot s^{-2}})$.

The following kinematic and dynamic parameters of the planar robot shown in Fig. 3.1 are used for numerical simulations: the lengths of links $a_1 = \mathrm{O_0O_1} = 0.4$ and $a_2 = \mathrm{O_1E} = 0.5\ (\mathrm{m})$; the masses of links $m_1 = 20$ and $m_2 = 10\ (\mathrm{kg})$;the positions of the centers of mass $x_{\mathrm{C}_1} = \mathrm{O_0C_1} = 0.2, x_{\mathrm{C}_2} = \mathrm{O_1C_2} = 0.25\ (\mathrm{m})$; and the inertia moments of links $I_1 = 0.2667$ and $I_2 = 0.2083\ (\mathrm{kg \cdot m^2})$. The inertia matrix of the motors (reflected through the square of the gear ratios) is $\boldsymbol{B} = \mathrm{diag}(4.0572,\ 2.2148)\ (\mathrm{kg \cdot m^2})$. Because the robot has two flexible revolute joints, $\boldsymbol{W}$ is the identity matrix.

In addition, the gain matrices of the PD controller are chosen as follows:

$$\boldsymbol{K}_{\mathrm{P}} = \mathrm{diag}(3600, 3600), \quad (\mathrm{N{\cdot}m/rad}) \tag{3.36}$$

$$\boldsymbol{K}_{\mathrm{D}} = \mathrm{diag}(60, 60) \quad (\mathrm{N{\cdot}m{\cdot}s/rad}) \tag{3.37}$$

For the direct eigenvalue problem, the stiffness and damping matrices are given by:

$$\boldsymbol{K} = \mathrm{diag}(2000, 1600), \quad (\mathrm{N{\cdot}m/rad}) \tag{3.38}$$

$$\boldsymbol{D} = \mathrm{diag}(3.91, 2.21), \quad (\mathrm{N{\cdot}m{\cdot}s/rad}) \tag{3.39}$$

$$\boldsymbol{D}_{\mathrm{m}} = \mathrm{diag}(2.04, 1.68) \quad (\mathrm{N{\cdot}m{\cdot}s/rad}) \tag{3.40}$$

Table 3.1 Simulated frequencies and damping ratios

	joint		modal data 1		modal data 2	
configuration	q_1(rad)	q_2(rad)	$\hat{f}_1$ (Hz)	$\hat{\zeta}_1$ (%)	$\hat{f}_4$ (Hz)	$\hat{\zeta}_4$ (%)
1	1.474	1.765	2.83	12.03	9.38	12.34
2	0.133	1.947	3.01	13.06	9.15	13.54
3	2.193	0.839	2.29	8.75	12.1	7.72
4	0.667	0.501	2.18	8.29	13.68	7.47
5	0.946	0.443	2.16	8.28	13.93	7.49
6	2.24	2.208	3.26	15.08	8.97	14.65
7	-1.373	1.717	2.93	11.5	9.46	11.98
8	0.849	0.179	2.13	8.1	14.88	7.62
9	0.549	2.582	3.53	17.7	8.87	15.46
10	2.945	2.577	3.58	17.65	8.88	15.33
11	-2.307	2.945	3.78	19.04	8.92	14.99
12	1.627	0.974	2.32	9.07	11.51	8.08
13	-2.088	1.638	2.88	11.08	9.61	11.39
14	2.335	1.686	2.81	11.54	9.52	11.75
15	1.856	2.96	3.66	19.03	8.95	14.57

3.5.2 Modal data

In the method presented here, the robot's modal parameters must be obtained first. Appendix II shows how the natural frequencies and damping ratios of the robot are computed from the

eigenvalues of the direct eigenvalue problem. To consider only partitioned modal parameters of the several modes estimated in practice, we use modal parameters in the first mode ($\hat{f}_1, \hat{\xi}_1$) and fourth mode ($\hat{f}_4, \hat{\xi}_4$) as the input of the optimization problem. In addition, to guarantee the constraint Eq. (3.32), we use 15 arbitrary configurations (see Fig. II-1 and Fig. II-4). The results are summarized in Table 3.1. It can be seen that the total number of natural frequencies/damping ratios is $N = 30$, while the number of unknowns is $\dim(\boldsymbol{k}) + \dim(\boldsymbol{d}) = 6$ (see Eq. (3.41)). Therefore, the condition in Eq. (3.32) is guaranteed.

Because this paper aims to use modal data to identify unknown parameters, the OMA tests on this real manipulator are unnecessary. However, to illustrate a real scenario, we suppose that the modal data are affected by noise. Because damping ratios are often difficult to estimate accurately, we add a small error to the damping ratios of the fifth and seventh columns in Table 3.1 in order to examine the influence of damping ratio deviation on the identified parameters. Note that the natural frequencies ($\hat{f}_1, \hat{f}_4$) are held constant.

The following cases are considered with different noise levels:

- Case 1: The values of the damping ratios ($\hat{\xi}_1, \hat{\xi}_4$) are held constant (the influence of the noise on the modal parameters is negligible).
- Cases 2 & 3: The values of the damping ratios ($\hat{\xi}_1, \hat{\xi}_4$) are modified by adding -0.5% and 0.5%, respectively.
- Cases 4 & 5: The values of the damping ratios ($\hat{\xi}_1, \hat{\xi}_4$) are modified by adding -1% and 1%, respectively.
- Cases 6 & 7: The values of the damping ratios ($\hat{\xi}_1, \hat{\xi}_4$) are modified by adding -2% and 2%, respectively.

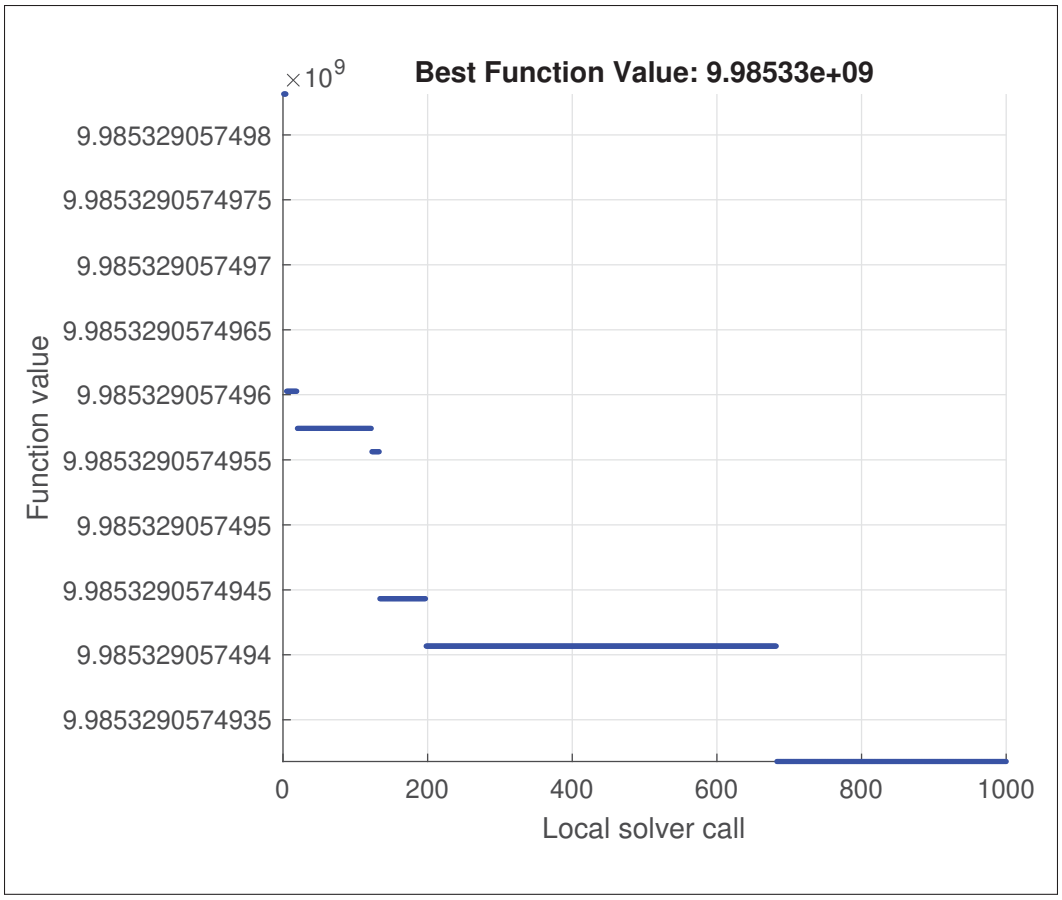

Figure 3.2 Local minimums

3.5.3 Results

The optimization problem presented in Section 3.4.2 is solved numerically using the MultiStart and lsqnonlin functions in Matlab to find $\boldsymbol{k}$ and $\boldsymbol{d}$ defined by

$$\boldsymbol{k} = [k_1, k_2]^\mathsf{T}, \quad \boldsymbol{d} = [d_{\mathrm{f},1}, d_{\mathrm{f},2}, d_{\mathrm{fm},1}, d_{\mathrm{fm},2}]^\mathsf{T} \tag{3.41}$$

with the bound constraints

$$0 \leq \boldsymbol{k} \leq 5000 \ (\mathrm{N{\cdot}m/rad}), \quad 0 \leq \boldsymbol{d} \leq 100 \ (\mathrm{N{\cdot}m{\cdot}s/rad}) \tag{3.42}$$

For example, the local minimums corresponding to the 1000 initial values of the second case are illustrated in Fig. 3.2.

Table 3.2 Identified stiffness and damping parameters of joints and motors

	k_1	k_2	$d_{\mathrm{f},1}$	$d_{\mathrm{f},2}$	$d_{\mathrm{fm},1}$	$d_{\mathrm{fm},2}$	$F(\boldsymbol{k}^\star, \boldsymbol{d}^\star)$
true value	2000	1600	3.91	2.21	2.04	1.68	0.0
Case 1	2000	1600	3.91	2.21	2.04	1.68	2.49e-11
Case 2	1997.6	1600.3	2.84	2.10	2.86	0.00	9.98e+09
Case 3	2002.3	1600.1	4.99	2.31	1.09	3.98	9.15e+09
Case 4	1995.0	1601.8	1.78	1.97	3.55	0.0	5.30e+10
Case 5	2004.9	1600.1	6.10	2.41	0.0	6.20	3.65e+10
Case 6	1994.0	1604.5	0.0	1.66	2.82	0.0	2.61e+11
Case 7	2005.4	1600.7	7.96	2.64	0.0	10.46	1.50e+11

The results obtained in all seven cases are summarized in Table 3.2, in which the last column describes the minimum of the objective function. It can be seen in case 1 that if modal parameters are not affected by noise, the stiffness and damping parameters are accurately identified. In addition, because the objective function is nearly equal to zero, the solution obtained in this case is the globally optimal solution. In the other cases, the obtained stiffness parameters are close to the 'true' values. However, the obtained damping parameters are significantly affected by the damping ratio deviation. For example, in cases 5 and 7, the damping parameters are unreliable, especially for parameters $d_{\mathrm{f},1}$ and $d_{\mathrm{fm},2}$, when the damping ratios are biased about 1% and 2%.

3.6 Conclusions and future work

In this paper, we have presented an optimization procedure for estimating the stiffness and damping parameters of flexible joint robots. The objective function was established based on the linearized dynamic model and measured modal information. We have proposed an optimization strategy based on multiple initial values to solve the optimization problem. Numerical simulations for a planar robot with two flexible joints showed that all stiffness and damping parameters could be accurately identified if natural frequencies and ratios were measured correctly. The effect of noise on identified parameters was investigated to illustrate the real-life modal data.

The proposed method requires identifying the robot's natural frequencies and damping ratios in several different configurations, and so the selection of natural frequencies and damping ratios in the frequency and damping ranges will be the subject of future work. The optimal number of modal parameters utilized to identify stiffness and damping parameters will also be studied further. We expect to apply the proposed method to real applications for robotic manipulators with multiple flexible joints.

CHAPTER 4

SYMBOLIC DIFFERENTIATION ALGORITHM FOR INVERSE DYNAMICS OF SERIAL ROBOTS WITH FLEXIBLE JOINTS

Thanh Trung Do[1] , Viet Hung Vu[1] , Zhaoheng Liu[1]

[1] Department of Mechanical Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper published in *Journal of Mechanisms and Robotics*, vol. 13, December 2021

Abstract

A new symbolic differentiation algorithm is proposed in this paper to automatically generate the inverse dynamics of flexible joint robots in symbolic form, and results obtained can be used in real-time applications. The proposed method with $O(n)$ computational complexity is developed based on the recursive Newton-Euler algorithm, the chain rule of differentiation, and the computer algebra system. The input of the proposed algorithm consists of symbolic matrices describing the kinematic and dynamic parameters of the robot. The output is the inverse dynamics solution written in portable and optimized code (C-code/Matlab-code). An exemplary, numerical simulation for inverse dynamics of the Kuka LWR4 robot with seven flexible joints is conducted using Matlab, in which the computational time per cycle of inverse dynamics is about 0.02 millisecond. The numerical example provides very good matching results versus existing methods, while requiring much less computation time and complexity.

4.1 Introduction

Recent years have seen a great deal of attention directed at dynamic modeling, identification, and control for lightweight robots with rigid links and flexible joints (Albu-Schäffer *et al.*, 2007; Moberg, 2010; Hazel *et al.*, 2012; Kim & Croft, 2018). The objectives are to increase the load to weight ratio, to achieve low energy consumption, to allow access to hard-to-reach areas, as well as to improve the safety when interacting with humans. In machining applications such as

milling, grinding, and polishing processes using industrial robots, the vibration reduction of the robot's links as well as tool center point due to changing high cutting/impact forces, and therefore, improve the surface accuracy, is the essential task (Iglesias *et al.*, 2015).

In order to reduce undesired vibration due to flexible transmission elements, model-based control methods such as feedforward+PD, static/dynamic feedback linearization, and passivity-based controls (Albu-Schäffer *et al.*, 2007; Chung *et al.*, 2016; De Luca & Book, 2016) are used to compute driving motor torques. It can be pointed out that the main component of any model-based controller is the inverse dynamics module, which is used to calculate motor torques in order to realize desired motions. For real-time applications, the computational efficiency of inverse dynamics is the most critical factor because the sampling time of 1 (ms) is often used in practice.

The inverse dynamics problem of robots with rigid joints and links has been intensively studied for over forty years. There are three main methods used to formulate equations of motion and compute the inverse dynamics: Newton-Euler, Lagrangian, and Kane formulations (Kane & Levinson, 1983; Schiehlen, 1990). These methods may be formulated in numerical/symbolical formalism, in non-recursive/recursive form (Schiehlen, 1997). Inverse dynamics based on non-recursive Newton-Euler formulation was addressed in Schiehlen & Erberhard (2014), while a non-recursive method based on the Lagrangian formulation can be found in any robotic textbook (Siciliano *et al.*, 2009). The recursive form of the Lagrangian formulation was proposed in Hollerbach (1980). The well-known $O(n)$ recursive Newton-Euler algorithm (RNEA) (Luh, Walker & Paul, 1980) is the most efficient algorithm used to compute the robot's inverse dynamics in real-time, especially when the number of links $n \geq 6$. The most compact form of RNEA was proposed in Featherstone (2008), where spatial notations are used to compute kinematic/dynamic quantities of rigid links. Inverse dynamics of redundantly actuated systems was presented in Wang, Belzile, Angeles & Li (2019) using natural orthogonal complement (NOC) and Moore-Penrose pseudoinverse methods. In Khalil, Boyer & Morsli (2017), a general algorithm based on RNEA was addressed to solve direct/inverse dynamics of rigid/flexible robots with tree structure having fixed or floating base. Recently, a general approach using the modular

form of RNEA is proposed in Kumar, Szadkowski, Mueller & Kirchner (2020) to solve inverse dynamics for series-parallel hybrid systems with closed loops. Several multibody software, such as SYMORO+ (Khalil & Creusot, 1997), MapleSim (DynaFlexPro) (Schmitke & Goossens, 2011), and Neweul-M^2 (Kurz, Eberhard, Henninger & Schiehlen, 2010), can be used to derive the inverse dynamics solution in symbolic form.

The inverse dynamics problem of flexible-joint robots is more complicated than that of rigid-joint robots since the number of actuators of a flexible joint robot is less than its number of degrees of freedom. In addition, higher-order derivatives of the joint positions are required to compute the second derivative of torques of rigid links (De Luca, 2000).

Recently, a very efficient algorithm with $O(n)$ complexity, called EJNEA, was proposed in Buondonno & De Luca (2015); Buondonno & De Luca (2016) to solve the inverse dynamics of flexible-joint robots in real-time. For the case of the Kuka LWR4 robot, the computation time per cycle using EJNEA takes approximately 0.033 (ms) in Matlab using the standard computer. From the results obtained in Guarino Lo Bianco (2009), authors in Buondonno & De Luca (2015) developed analytical formulations to compute higher-order derivatives of kinematic/dynamic quantities recursively. This algorithm can be used to design model-based controllers for robot systems with series elastic actuators (SEA) (Pratt & Williamson, 1995) as well as with variable stiffness actuators (VSA) (Buondonno & De Luca, 2016). A simple version of EJNEA was applied for the passivity-based tracking controller (Giusti *et al.*, 2018).

Based on the screw/Lie group formulation (Müller, 2017), an $O(n)$ algorithm was proposed to compute the first and second derivatives of inverse dynamics for serial manipulators. The similar approach was discussed thoroughly in Park, Kim, Jang & Hong (2018). Transformation matrices for the kinematic description are established using the product of exponential formulation. Kinematic and dynamic quantities of links are computed using Lie brackets of joint screws. The advantage of this approach is that the inverse dynamics solution of rigid links and its time derivatives can be expressed in compact form. In Yang *et al.* (2018), a new $O(n)$ recursive algorithm using Lie groups, called RLGA, was presented. This algorithm was applied to

compute driving torques of heavy-duty industrial robots in the robotic friction stir welding process. The number of operations of RLGA is even less than that of EJNEA, i.e., the former provides better performance for real-time applications.

In this work, a new algorithm with $O(n)$ complexity, called SDRNEA, is presented to automatically generate inverse dynamics of serial robots with flexible joints in symbolic form. It was inspired by the work (Buondonno & De Luca, 2015). The proposed algorithm based on computer algebra systems is combined with two symbolic algorithms featuring optimized code generation. To the best of our knowledge, this is the first study in which the inverse dynamics solution of flexible-joint robots has been completed in symbolic form.

This paper is organized as follows. In Sect. 4.2, the dynamic modeling and inverse dynamics of flexible-joint robots in reduced form are briefly addressed. Then, the mathematical background is presented in Sect. 4.3. In Sect. 4.4, the proposed algorithm is discussed to compute torques of rigid links and their derivatives in symbolic form. In Sect. 4.5, the inverse dynamics problem of the Kuka LWR4 robot with seven flexible joints is simulated using Matlab. Finally, conclusions are given in Sect. 4.6.

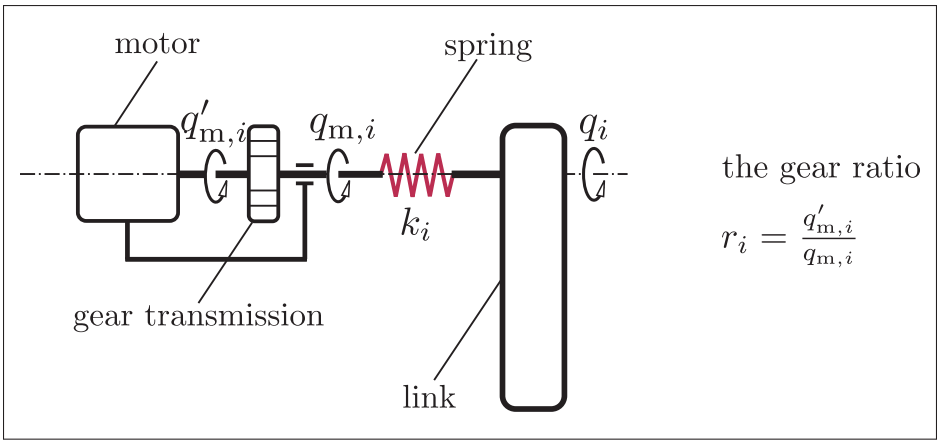

Figure 4.1 Model of a flexible joint

4.2 Dynamic equations and inverse dynamics

This section presents the dynamic modeling and inverse dynamics problem of flexible-joint robots using the assumptions presented in De Luca & Book (2016). Because the Spong's assumption (high gear ratios of motors, > 100) is used equations of motion is expressed in reduced form (Spong, 1987). Note that equations of motion in completed form were presented in Tomei (1990); De Luca & Tomei (1996); De Luca & Book (2016).

4.2.1 Dynamic equations

Each flexible joint is described by two coordinates as shown in Fig. 4.1: joint coordinate q_i and motor coordinate $q_{\mathrm{m},i}$. Note that $q'_{\mathrm{m},i}$ is the motor coordinate before the reduction gears with which $q'_{\mathrm{m},i} = r_i q_{\mathrm{m},i}$ and the gear ratio r_i.

The robot with n flexible joints is modeled as $2n$ rigid bodies: n rigid links and n rotors. The robot's configuration is described by two vectors of joint and motor coordinates: $\boldsymbol{q} = [q_1,\, q_2, \ldots,\, q_n]^\mathsf{T}$ and $\boldsymbol{q}_\mathrm{m} = \left[q_{\mathrm{m},1},\, q_{\mathrm{m},2}, \ldots,\, q_{\mathrm{m},n}\right]^\mathsf{T}$. Using the Lagrangian formulation and Spong's assumption, equations of motion of the robot in reduced form (neglecting the inertial couplings between links and motors) are expressed as (Spong, 1987; De Luca & Tomei, 1996; De Luca & Book, 2016):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_\mathrm{f}(\dot{\boldsymbol{q}}) = \boldsymbol{K}(\boldsymbol{q}_\mathrm{m} - \boldsymbol{q}) \tag{4.1}$$

$$\boldsymbol{J}_\mathrm{m}\ddot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{\tau}_\mathrm{fm}(\dot{\boldsymbol{q}}_\mathrm{m}) + \boldsymbol{K}(\boldsymbol{q}_\mathrm{m} - \boldsymbol{q}) = \boldsymbol{\tau}_\mathrm{m} \tag{4.2}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is a mass matrix of rigid links (including the mass of motors); $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is a Coriolis matrix; $\boldsymbol{g} \in \mathbb{R}^n$ is a vector of gravity forces; $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a constant diagonal matrix of spring constants; $\boldsymbol{K}(\boldsymbol{q} - \boldsymbol{q}_\mathrm{m}) \in \mathbb{R}^n$ is a vector of elastic torques; $\boldsymbol{J}_\mathrm{m} \in \mathbb{R}^{n\times n}$ is a constant diagonal matrix of the rotor and gear inertia moments in their rotational axes; $\boldsymbol{\tau}_\mathrm{m} \in \mathbb{R}^n$ is a vector of motor torques; $\boldsymbol{\tau}_\mathrm{f},\, \boldsymbol{\tau}_\mathrm{fm} \in \mathbb{R}^n$ are vectors of joint and motor frictions, and they are modeled

as (Armstrong-Helouvry, 1991):

$$\boldsymbol{\tau}_{\mathrm{f}} = \boldsymbol{D}_{\mathrm{v}}\dot{\boldsymbol{q}} + \boldsymbol{D}_{\mathrm{c}}\mathrm{sign}(\dot{\boldsymbol{q}}) \tag{4.3}$$

$$\boldsymbol{\tau}_{\mathrm{fm}} = \boldsymbol{D}_{\mathrm{v,m}}\dot{\boldsymbol{q}}_{\mathrm{m}} + \boldsymbol{D}_{\mathrm{c,m}}\mathrm{sign}(\dot{\boldsymbol{q}}_{\mathrm{m}}) \tag{4.4}$$

where $\boldsymbol{D}_{\mathrm{v}},\ \boldsymbol{D}_{\mathrm{v,m}} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D}_{\mathrm{c}},\ \boldsymbol{D}_{\mathrm{c,m}} \in \mathbb{R}^{n\times n}$ are the diagonal matrices of viscous and Coulomb damping coefficients, respectively. Note that the effect of external forces/moments and internal damping of flexible joints is not considered in this work.

Eq. (4.1) and Eq. (4.2) describes the dynamics of rigid links and motors, respectively. If all joints are rigid, then $\boldsymbol{q}_{\mathrm{m}} = \boldsymbol{q}$, one can therefore combine Eq. (4.1) and Eq. (4.2) into a single equation:

$$(\boldsymbol{M}(\boldsymbol{q}) + \boldsymbol{J}_{\mathrm{m}})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{\mathrm{f}}^{\star}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau}_{\mathrm{m}} \tag{4.5}$$

where $\boldsymbol{\tau}_{\mathrm{f}}^{\star}$ is a vector of joint and motor frictions.

4.2.2 Inverse dynamics

If all joints are rigid, $\boldsymbol{\tau}_{\mathrm{m}}$ can be directly computed from Eq. (4.5) for the desired joint motions $\boldsymbol{q}^{(\mathrm{d})}, \dot{\boldsymbol{q}}^{(\mathrm{d})}$, and $\ddot{\boldsymbol{q}}^{(\mathrm{d})}$. For the flexible-joint robot, $\boldsymbol{\tau}_{\mathrm{m}}$ is computed using Eq. (4.1) and Eq. (4.2) (De Luca, 2000).

Indeed, by denoting $\boldsymbol{\tau}$ as a vector of driving torques of rigid links:

$$\boldsymbol{\tau} \triangleq \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\ \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{4.6}$$

$\boldsymbol{q}_{\mathrm{m}}$ is found from Eq. (4.1) as follows:

$$\boldsymbol{q}_{\mathrm{m}} = \boldsymbol{q} + \boldsymbol{K}^{-1}(\boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{f}}) \tag{4.7}$$

Thus, the first and second time derivatives of $\boldsymbol{q}_\mathrm{m}$ are:

$$\dot{\boldsymbol{q}}_\mathrm{m} = \dot{\boldsymbol{q}} + \boldsymbol{K}^{-1}(\dot{\boldsymbol{\tau}} + \dot{\boldsymbol{\tau}}_\mathrm{f}) = \dot{\boldsymbol{q}} + \boldsymbol{K}^{-1}(\dot{\boldsymbol{\tau}} + \boldsymbol{D}_\mathrm{v}\ddot{\boldsymbol{q}}) \tag{4.8}$$

$$\ddot{\boldsymbol{q}}_\mathrm{m} = \ddot{\boldsymbol{q}} + \boldsymbol{K}^{-1}(\ddot{\boldsymbol{\tau}} + \ddot{\boldsymbol{\tau}}_\mathrm{f}) = \ddot{\boldsymbol{q}} + \boldsymbol{K}^{-1}(\ddot{\boldsymbol{\tau}} + \boldsymbol{D}_\mathrm{v}\dddot{\boldsymbol{q}}) \tag{4.9}$$

where $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}} \in \mathbb{R}^n$ denote are the first and second time derivatives of $\boldsymbol{\tau}$, respectively. Substituting Eq. (4.7), Eq. (4.8) and Eq. (4.9) into Eq. (4.2), the vector of motor torques $\boldsymbol{\tau}_\mathrm{m}$ is obtained as:

$$\boldsymbol{\tau}_\mathrm{m} = \boldsymbol{J}_\mathrm{m}\ddot{\boldsymbol{q}}_\mathrm{m} + \boldsymbol{\tau}_\mathrm{fm}(\dot{\boldsymbol{q}}_\mathrm{m}) + \boldsymbol{\tau} \tag{4.10}$$

As a result, given the desired motion in the joint space such as $\boldsymbol{q}^{(\mathrm{d})}$, $\dot{\boldsymbol{q}}^{(\mathrm{d})}$, $\ddot{\boldsymbol{q}}^{(\mathrm{d})}$, $\dddot{\boldsymbol{q}}^{(\mathrm{d})}$ and $\ddddot{\boldsymbol{q}}^{(\mathrm{d})}$, the inverse dynamics solution of the flexible joint robot is determined by Eq. (4.10) where $\dot{\boldsymbol{q}}_\mathrm{m}^{(\mathrm{d})}$ and $\ddot{\boldsymbol{q}}_\mathrm{m}^{(\mathrm{d})}$ are determined by Eq. (4.8) and Eq. (4.9).

For a flexible-joint robot, the most challenging of inverse dynamics in real-time applications is how to formulate $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$, and $\ddot{\boldsymbol{\tau}}$ efficiently. If $\boldsymbol{\tau}$ is derived using the Lagrangian formulation, the computation of $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ is extremely complicated. A new symbolic algorithm is presented in Sect. 4.4 to compute $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ efficiently using computer algebra systems.

4.3 Mathematical background

In this section, several essential notations are presented for computing inverse dynamics of robots.

4.3.1 Kinematic representation

Consider two rigid links, $i-1$ and i, in the inertial reference frame $O_0x_0y_0z_0$, connected by a revolute/translational joint i shown in Fig. 4.2. Two body-fixed frames $(\mathrm{BF})_{i-1}$ $(\mathrm{O}_{i-1}x_{i-1}y_{i-1}z_{i-1})$ and $(\mathrm{BF})_i$ $(\mathrm{O}_i x_i y_i z_i)$ are defined using the Denavit-Hartenberg (DH) convention (Siciliano *et al.*,

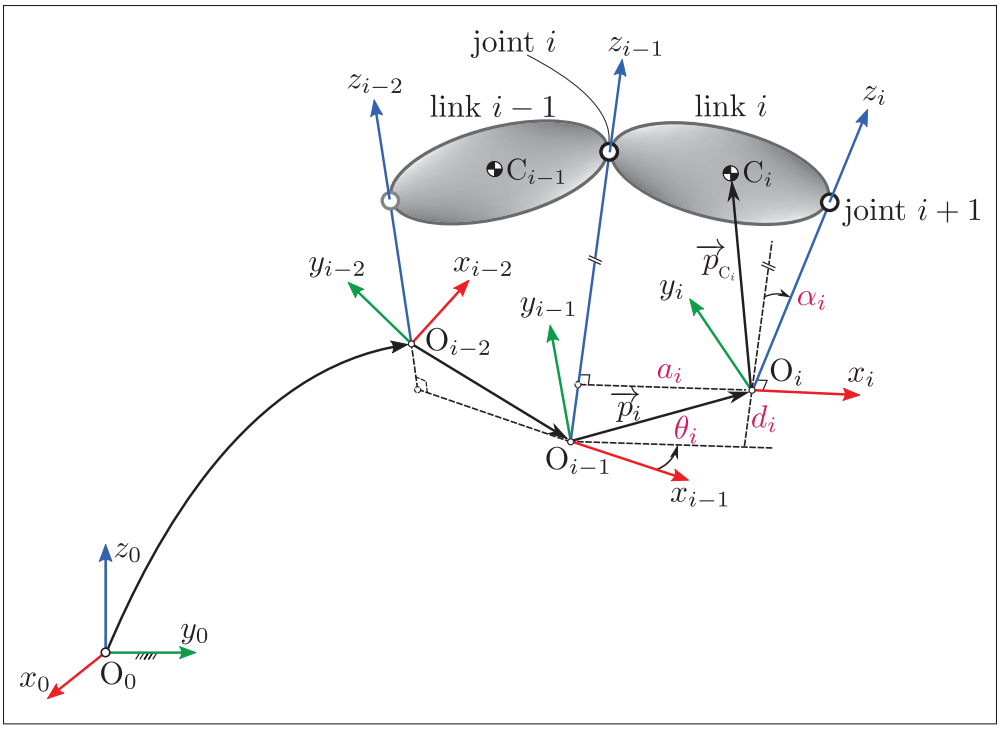

Figure 4.2 Kinematic representations of two links using DH convention

2009). Therefore, the position of O_i and orientation of link i in $(BF)_{i-1}$ are expressed as:

$$\boldsymbol{T}_i = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{p}_i \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{4.11}$$

where $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ is the rotation matrix and $\boldsymbol{p}_i \in \mathbb{R}^3$ is the position vector of O_i with respect to O_{i-1}. Using four DH parameters $(\theta_i, d_i, a_i, \alpha_i)$, the terms in Eq. (4.11) are given by:

$$\boldsymbol{p}_i = \begin{bmatrix} a_i \cos\theta_i, & a_i \sin\theta_i, & d_i \end{bmatrix}^{\mathsf{T}}, \tag{4.12}$$

$$\boldsymbol{R}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{4.13}$$

Since $(\boldsymbol{R}_i)^{-1} = \boldsymbol{R}_i^{\mathsf{T}}$, the position of O_i with respect to O_{i-1} expressed in $(\mathrm{BF})_i$ is defined by:

$$\boldsymbol{p}_i^{\star} = \boldsymbol{R}_i^{\mathsf{T}} \boldsymbol{p}_i = \left[a_i,\ d_i \sin\alpha_i,\ d_i \cos\alpha_i\right]^{\mathsf{T}} \tag{4.14}$$

4.3.2 Kinematic and dynamic parameters

For a robot with n rigid links, DH parameters can be grouped by a matrix $\boldsymbol{H} \in \mathbb{R}^{n\times 5}$ as:

$$\boldsymbol{H} = \begin{bmatrix} \theta_1 & d_1 & a_1 & \alpha_1 & \xi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_n & d_n & a_n & \alpha_n & \xi_n \end{bmatrix} \tag{4.15}$$

where $\theta_i = q_i$ and $\xi_i = 0$ for a prismatic joint, while $d_i = q_i$ and $\xi_i = 1$ for a revolute joint, $i = (1, \ldots, n)$.

Dynamic parameters of the link i are assumed to be constant including: m_i is the mass; $\boldsymbol{p}_{\mathrm{c},i} = [x_{\mathrm{c},i}, y_{\mathrm{c},i}, z_{\mathrm{c},i}]^{\mathsf{T}}$ is the position vector of the center of mass C_i with respect to O_i in $(\mathrm{BF})_i$; and $\boldsymbol{I}_{\mathrm{c},i}$ is the inertia tensor of this link with respect to C_i in $(\mathrm{BF})_i$. Because $\boldsymbol{I}_{\mathrm{c},i}$ has six independent components ($I_{\mathrm{xx},i}$, $I_{\mathrm{xy},i}$, $I_{\mathrm{xz},i}$, $I_{\mathrm{yy},i}$, $I_{\mathrm{yz},i}$, $I_{\mathrm{zz},i}$), dynamic parameters of n links can be grouped by a matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times 10}$ as:

$$\boldsymbol{\Gamma} = \begin{bmatrix} m_1 & x_{\mathrm{c},1} & y_{\mathrm{c},1} & z_{\mathrm{c},1} & I_{\mathrm{xx},1} & I_{\mathrm{xy},1} & I_{\mathrm{xz},1} & I_{\mathrm{yy},1} & I_{\mathrm{yz},1} & I_{\mathrm{zz},1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_n & x_{\mathrm{c},n} & y_{\mathrm{c},n} & z_{\mathrm{c},n} & I_{\mathrm{xx,n}} & I_{\mathrm{xy,n}} & I_{\mathrm{xz,n}} & I_{\mathrm{yy,n}} & I_{\mathrm{yz,n}} & I_{\mathrm{zz,n}} \end{bmatrix} \tag{4.16}$$

4.4 Symbolic derivatives for inverse dynamics

In this section, symbolic algorithms are presented to establish $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$, and $\ddot{\boldsymbol{\tau}}$ in symbolic form. The following notations are used:

- $\boldsymbol{g}_0 \in \mathbb{R}^3$: the gravity vector,
- $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \dddot{\boldsymbol{q}}$ and $\ddddot{\boldsymbol{q}} \in \mathbb{R}^n$: vectors of joint positions, velocities, accelerations, jerks, and snaps,
- $\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i \in \mathbb{R}^3$: vectors of angular velocities and accelerations of link i,
- $\boldsymbol{a}_i, \boldsymbol{a}_{\mathrm{c},i} \in \mathbb{R}^3$: vectors of linear accelerations of O_i and C_i,
- $\boldsymbol{p}_{\mathrm{c},i} \in \mathbb{R}^3$: position vector of C_i in $(\mathrm{BF})_i$,
- $\boldsymbol{f}_{\mathrm{c},i} \in \mathbb{R}^3$: vector of inertial forces at C_i, and $\boldsymbol{f}_i, \boldsymbol{n}_i \in \mathbb{R}^3$: vectors of reaction forces and moments exerted on link i by link $i-1$,
- $\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{19n}$: vectors of intermediate and torque variables, and their derivatives,
- $\boldsymbol{S} \in \mathbb{R}^{19n\times 2}$ and $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{16n\times 2}$: symbolic matrices including intermediate variables and expressions.

Note that all kinematic/dynamic quantities of link i are expressed in its frame $(\mathrm{BF})_i$. In addition, a skew-symmetric matrix of a vector $\boldsymbol{b} \in \mathbb{R}^3$ is denoted by $\widehat{\boldsymbol{b}}$.

4.4.1 Symbolic recursive Newton-Euler algorithm

The symbolic recursive Newton-Euler algorithm with $O(n)$ complexity, called SRNEA, is proposed in Alg. 1 to automatically generate $\boldsymbol{\tau}$ in symbolic form. This algorithm is developed based on RNEA using the numerical formalism (Luh *et al.*, 1980; Siciliano *et al.*, 2009), and symbolic formalism (Khalil & Dombre, 2004). In comparison to RNEA in Khalil & Dombre (2004), our method is more straightforward and systematic.

The foundational idea of SRNEA is the use of intermediate variables to store and replace intermediate expressions during the forward and backward recursive computations. Therefore, the recursive computation with two loops using the numerical formalism is entirely replaced by the computation sequence without loops.

In Alg. 1, the forward recursion used to compute kinematics from link 1 to link n includes three stages: (1) recursive angular velocities, (2) recursive angular accelerations, and (3) recursive linear accelerations. This strategy allows us to extract angular acceleration components of all links efficiently, and use them to compute $\dot{\boldsymbol{\tau}}$ (see Alg. 2, from line 5 to line 7). The backward

Algorithm 1 Recursive Newton-Euler algorithm using the symbolic approach

Inputs:
$\boldsymbol{H}$, $\boldsymbol{\Gamma}$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\boldsymbol{g}_0$, $\boldsymbol{s}$
Outputs:
$\boldsymbol{S}$
Initialize:
$k = 0$, $\boldsymbol{S} = \boldsymbol{0}$, $n = \mathsf{Dim}(\boldsymbol{q})$, $\boldsymbol{\omega}_0 = \boldsymbol{0}$, $\dot{\boldsymbol{\omega}}_0 = \boldsymbol{0}$, $\boldsymbol{a}_0 = -\boldsymbol{g}_0$, $\boldsymbol{f}_{n+1} = \boldsymbol{0}$, $\boldsymbol{n}_{n+1} = \boldsymbol{0}$,
$\boldsymbol{z}_0 = [0, 0, 1]^\mathsf{T}$, $\boldsymbol{R}_{n+1} = \mathrm{diag}(1, 1, 1)$
for $i = 1$ **by** 1 **to** n **do** ▷ the forward recursion
  $[\boldsymbol{R}_i, \boldsymbol{p}_i^\star, \xi_i] = \texttt{GetDHParameters}(\boldsymbol{H}, i)$
  $\boldsymbol{z}_i = \boldsymbol{R}_i^\mathsf{T} \boldsymbol{z}_0$
  $\boldsymbol{\omega}_i = \boldsymbol{R}_i^\mathsf{T} (\boldsymbol{\omega}_{i-1} + \xi_i \dot{q}_i \boldsymbol{z}_0)$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+1}, s_{k+2}, s_{k+3}], \boldsymbol{\omega}_i)$
  $\boldsymbol{\omega}_i = \texttt{ReplaceExpressions}(\boldsymbol{\omega}_i, [s_{k+1}, s_{k+2}, s_{k+3}])$
  $k = k + 3$
end for
for $i = 1$ **by** 1 **to** n **do**
  $\dot{\boldsymbol{\omega}}_i = \boldsymbol{R}_i^\mathsf{T} (\dot{\boldsymbol{\omega}}_{i-1} + \xi_i (\ddot{q}_i \boldsymbol{z}_0 + \dot{q}_i \widehat{\boldsymbol{\omega}}_{i-1} \boldsymbol{z}_0))$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+1}, s_{k+2}, s_{k+3}], \dot{\boldsymbol{\omega}}_i)$
  $\dot{\boldsymbol{\omega}}_i = \texttt{ReplaceExpressions}(\dot{\boldsymbol{\omega}}_i, [s_{k+1}, s_{k+2}, s_{k+3}])$
  $k = k + 3$
end for
for $i = 1$ **by** 1 **to** n **do**
  $\boldsymbol{\Omega}_i = \widehat{\dot{\boldsymbol{\omega}}}_i + \widehat{\boldsymbol{\omega}}_i^2$
  $\boldsymbol{a}_i = \boldsymbol{R}_i^\mathsf{T} \boldsymbol{a}_{i-1} + \boldsymbol{\Omega}_i \boldsymbol{p}_i^\star + (1 - \xi_i)(\ddot{q}_i \boldsymbol{z}_i + 2 \dot{q}_i \widehat{\boldsymbol{\omega}}_i \boldsymbol{z}_i)$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+1}, s_{k+2}, s_{k+3}], \boldsymbol{a}_i)$
  $\boldsymbol{a}_i = \texttt{ReplaceExpressions}(\boldsymbol{a}_i, [s_{k+1}, s_{k+2}, s_{k+3}])$
  $k = k + 3$
end for
for $i = n$ **by** -1 **to** 1 **do** ▷ the backward recursion
  $[m_i, \boldsymbol{p}_{\mathrm{c},i}, \boldsymbol{I}_i] = \texttt{GetMCIParameters}(\boldsymbol{\Gamma}, i)$
  $\boldsymbol{f}_{\mathrm{c},i} = m_i (\boldsymbol{a}_i + \boldsymbol{\Omega}_i \boldsymbol{p}_{\mathrm{c},i})$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+1}, s_{k+2}, s_{k+3}], \boldsymbol{f}_{\mathrm{c},i})$
  $\boldsymbol{f}_{\mathrm{c},i} = \texttt{ReplaceExpressions}(\boldsymbol{f}_{\mathrm{c},i}, [s_{k+1}, s_{k+2}, s_{k+3}])$
  $\boldsymbol{f}_i = \boldsymbol{R}_{i+1} \boldsymbol{f}_{i+1} + \boldsymbol{f}_{\mathrm{c},i}$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+4}, s_{k+5}, s_{k+6}], \boldsymbol{f}_i)$
  $\boldsymbol{f}_i = \texttt{ReplaceExpressions}(\boldsymbol{f}_i, [s_{k+4}, s_{k+5}, s_{k+6}])$
  $\boldsymbol{n}_i^\star = \boldsymbol{I}_i \dot{\boldsymbol{\omega}}_i + \widehat{\boldsymbol{\omega}}_i \boldsymbol{I}_i \boldsymbol{\omega}_i$
  $\boldsymbol{n}_i = \boldsymbol{R}_{i+1} \boldsymbol{n}_{i+1} + \widehat{\boldsymbol{p}}_i^\star \boldsymbol{f}_i + m_i \widehat{\boldsymbol{p}}_{\mathrm{c},i} \boldsymbol{a}_i + \boldsymbol{n}_i^\star$
  $\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{k+7}, s_{k+8}, s_{k+9}], \boldsymbol{n}_i)$
  $\boldsymbol{n}_i = \texttt{ReplaceExpressions}(\boldsymbol{n}_i, [s_{k+7}, s_{k+8}, s_{k+9}])$
  $\tau_i = (1 - \xi_i) \boldsymbol{z}_i^\mathsf{T} \boldsymbol{f}_i + \xi_i \boldsymbol{z}_i^\mathsf{T} \boldsymbol{n}_i$
  $k = k + 9$
end for
$\boldsymbol{S} = \texttt{SaveExpressions}(\boldsymbol{S}, [s_{18n+1}, s_{18n+2}, \ldots, s_{19n}], [\tau_1, \tau_2, \ldots, \tau_n])$

recursion is used to compute inertial forces/moments, reaction forces/moments and joint torques from link n to link 1.

The input vector $\boldsymbol{s}$ of SRNEA composes of $19n$ intermediate variables which is used to store and replace all intermediate expressions in recursive computations. Note that the last n elements of $\boldsymbol{s}$ represents n variables of $\boldsymbol{\tau}$. The output of SRNEA is the symbolic matrix $\boldsymbol{S}$ in which the first and second column are used to store intermediate variables and expressions, respectively. Generally, the structure of $\boldsymbol{S}$ is given by:

$$\boldsymbol{S} = \left[\boldsymbol{S}_{\omega}^{\mathsf{T}},\ \boldsymbol{S}_{\dot{\omega}}^{\mathsf{T}},\ \boldsymbol{S}_{\mathrm{a}}^{\mathsf{T}},\ \boldsymbol{S}_{\mathrm{br}}^{\mathsf{T}},\ \boldsymbol{S}_{\tau}^{\mathsf{T}}\right]^{\mathsf{T}} \tag{4.17}$$

where $\boldsymbol{S}_{\omega}, \boldsymbol{S}_{\dot{\omega}}, \boldsymbol{S}_{\mathrm{a}} \in \mathbb{R}^{3n\times 2}$ are matrices of angular velocities, angular accelerations and linear accelerations, respectively; $\boldsymbol{S}_{\mathrm{br}} \in \mathbb{R}^{9n\times 2}$ is a matrix of inertial forces/moments and reaction forces/moments. Note that $\boldsymbol{S}_{\tau} \in \mathbb{R}^{n\times 2}$ is a matrix of joint torques (see Alg. 1, line 40).

Four following functions are implemented in Alg. 1 to manipulate symbolic expressions.

- `GetDHParameters`$(\boldsymbol{H}, i)$ is used to extract kinematic parameters of link i from $\boldsymbol{H}$.
- `GetMCIParameters`$(\boldsymbol{\Gamma}, i)$ is used to extract dynamic parameters of link i from $\boldsymbol{\Gamma}$, in which the third output parameter is defined by: $\boldsymbol{I}_i = \boldsymbol{I}_{\mathrm{c},i} - m_i\widehat{\boldsymbol{p}}_{\mathrm{c},i}^2$.
- `SaveExpressions`(A, B, C) is used to save sequentially kinematic/dynamic quantities. Here, intermediate variables (B) and intermediate expressions (C) are saved in the first and second columns of A, respectively.
- `ReplaceExpressions`(C, B) is used to replace complex expressions (C) by intermediate variables (B).

4.4.2 Derivatives of recursive Newton-Euler algorithm

Given the symbolic matrix $\boldsymbol{S}$ obtained from Alg. 1, the row i of $\boldsymbol{S}$ may be expressed as:

$$s_i = \underbrace{S_{i,2}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, s_1, \ldots, s_{i-1})}_{f_i} \tag{4.18}$$

where s_i and f_i represent the intermediate variable and expression, respectively. Because f_i is an implicit function of time, but depends on $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, and $s_1, \ldots, s_{i-1}$, $\dot{f}_i$ is computed using the chain rule:

$$\begin{aligned}\dot{s}_i = \frac{\mathrm{d}f_i}{\mathrm{d}t} &= \underbrace{\frac{\partial f_i}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial f_i}{\partial \dot{\boldsymbol{q}}}\ddot{\boldsymbol{q}} + \frac{\partial f_i}{\partial \ddot{\boldsymbol{q}}}\dddot{\boldsymbol{q}}}_{\dot{f}_i^{\star}} + \sum_{k=1}^{i-1}\frac{\partial f_i}{\partial s_k}\dot{s}_k \\ &= \underbrace{\dot{f}_i^{\star} + \sum_{k=1}^{i-1}\frac{\partial f_i}{\partial s_k}\dot{s}_k}_{\dot{f}_i}\end{aligned} \tag{4.19}$$

Due to the special structure of $\boldsymbol{S}$ defined by Eq. (4.17), one has:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{S}_{\omega}) \equiv \boldsymbol{S}_{\dot{\omega}} \tag{4.20}$$

Therefore, Eq. (4.19) becomes:

$$\dot{s}_i = \begin{cases} s_{3n+i}, & \text{for } 1 \le i \le 3n \\ \dot{f}_i, & \text{for } 3n+1 \le i \le 19n \end{cases} \tag{4.21}$$

By taking time derivative both sides of Eq. (4.21), one obtains:

$$\ddot{s}_i = \begin{cases} \dot{s}_{3n+i}, & \text{for } 1 \le i \le 3n \\ \ddot{f}_i, & \text{for } 3n+1 \le i \le 19n \end{cases} \tag{4.22}$$

where $\ddot{f}_i$ is computed from Eq. (4.19):

$$\ddot{f}_i = \ddot{f}_i^{\star} + \sum_{k=1}^{i-1}\frac{\partial \dot{f}_i}{\partial s_k}\dot{s}_k + \sum_{k=1}^{i-1}\frac{\partial \dot{f}_i}{\partial \dot{s}_k}\ddot{s}_k \tag{4.23}$$

and $\ddot{f}_i^{\star}$ is defined as:

$$\ddot{f}_i^{\star} = \frac{\partial \dot{f}_i}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \dot{f}_i}{\partial \dot{\boldsymbol{q}}}\ddot{\boldsymbol{q}} + \frac{\partial \dot{f}_i}{\partial \ddot{\boldsymbol{q}}}\dddot{\boldsymbol{q}} + \frac{\partial \dot{f}_i}{\partial \dddot{\boldsymbol{q}}}\ddddot{\boldsymbol{q}} \tag{4.24}$$

From Eq. (4.21) and Eq. (4.22), a new algorithm, called DRNEA, is proposed in Alg. 2 to formulate $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ in symbolic form. The input of DRNEA is the symbolic matrix $\boldsymbol{S}$ obtained from SRNEA, while the output includes symbolic matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ composed of intermediate variables and expressions for evaluating $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$, respectively.

In Alg. 2, we use two vectors of intermediate variables, $\boldsymbol{u}$ and $\boldsymbol{v}$, to sequentially replace two vectors $\dot{\boldsymbol{s}}$ and $\ddot{\boldsymbol{s}}$ appearing on both sides of Eq. (4.21) and Eq. (4.22). This technique, also applied in Alg. 1, is straightforward to implement and computationally efficient. It allows reducing the complexity of resulting expressions when computing the derivative of large expressions.

As a result, the symbolic expressions used to compute $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ are derived from the matrices $\boldsymbol{S}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ as follows:

$$\begin{cases} S_{1,1} = S_{1,2} \\ \vdots \quad\quad \vdots \\ S_{19n,1} = S_{19n,2} \\ U_{1,1} = U_{1,2} \\ \vdots \quad\quad \vdots \\ U_{16n,1} = U_{16n,2} \\ V_{1,1} = V_{1,2} \\ \vdots \quad\quad \vdots \\ V_{16n,1} = V_{16n,2} \end{cases} \tag{4.25}$$

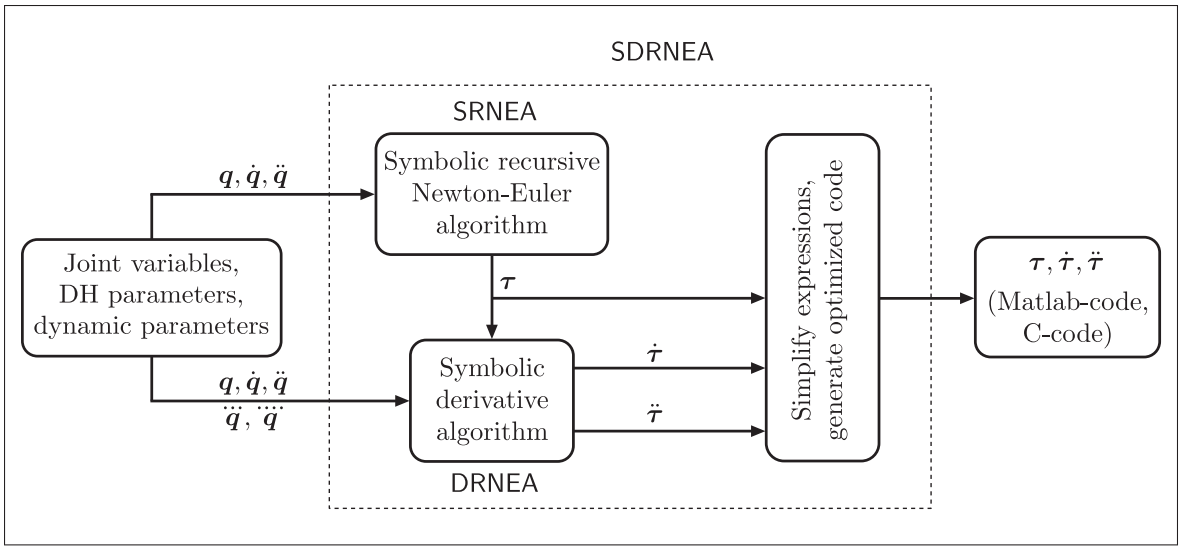

Figure 4.3 Schematic diagram of the proposed algorithm

4.4.3 Generation of optimized code

The direct use of Eq. (4.25) for evaluating $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ may be inefficient in real-time applications. This is because there are many common subexpressions in Eq. (4.25), which leads to redundant computation. This issue is briefly summarized as follows (Samin & Fisette, 2003; Uchida, 2011):

- Common subexpressions, such as trigonometric functions cos() and sin(), and products of these functions, are repeated many times.
- Constant subexpressions should be computed off-line.
- Several intermediate expressions are zero, redundant or useless due to special kinematic/dynamic parameters of the robot. These expressions should be eliminated.
- The intermediate variables have to be declared in the final output code. It is inefficient memory management if the number of variables are too large.

In Fig. 4.3, a framework is presented to generate optimized code using computer algebra systems. For example, the codegen[optimize] function in Maple can be used for this purpose (Wittkopf, 2008). Common subexpressions in Eq. (4.25) are identified and replaced automatically by new

Algorithm 2 Derivatives of recursive Newton-Euler algorithm using symbolic approach

Inputs:
$\boldsymbol{S}$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\dddot{\boldsymbol{q}}$, $\ddddot{\boldsymbol{q}}$, $\boldsymbol{s}$, $\boldsymbol{u}$, $\boldsymbol{v}$
Outputs:
$\boldsymbol{U}$, $\boldsymbol{V}$
Initialize:
$\boldsymbol{U} = \boldsymbol{0}$, $\boldsymbol{V} = \boldsymbol{0}$, $n = \mathsf{Dim}(\boldsymbol{q})$,
$\boldsymbol{q}^{\star} = [\boldsymbol{q}^{\mathsf{T}}, \dot{\boldsymbol{q}}^{\mathsf{T}}, \ddot{\boldsymbol{q}}^{\mathsf{T}}, \dddot{\boldsymbol{q}}^{\mathsf{T}}]^{\mathsf{T}}$,
$\dot{\boldsymbol{q}}^{\star} = [\dot{\boldsymbol{q}}^{\mathsf{T}}, \ddot{\boldsymbol{q}}^{\mathsf{T}}, \dddot{\boldsymbol{q}}^{\mathsf{T}}, \ddddot{\boldsymbol{q}}^{\mathsf{T}}]^{\mathsf{T}}$
for $i = 1$ **by** 1 **to** $19n$ **do** ▷ the first time derivative
  if $i <= 3n$ **then**
    $u_i = s_{3n+i}$
  else
    $f_i = S_{i,2}$
    $\dot{f}_i = \sum_{k=1}^{3n} \frac{\partial f_i}{\partial q_k^{\star}} \dot{q}_k^{\star} + \sum_{k=1}^{i-1} \frac{\partial f_i}{\partial s_k} u_k$
    $\boldsymbol{U} = \texttt{SaveExpressions}(\boldsymbol{U}, u_i, \dot{f}_i)$
  end if
end for
for $i = 1$ **by** 1 **to** $19n$ **do** ▷ the second time derivative
  if $i <= 3n$ **then**
    $v_i = u_{3n+i}$
  else
    $\ddot{f}_i = \sum_{k=1}^{4n} \frac{\partial \dot{f}_i}{\partial q_k^{\star}} \dot{q}_k^{\star} + \sum_{k=1}^{i-1} \frac{\partial \dot{f}_i}{\partial s_k} u_k + \sum_{k=1}^{i-1} \frac{\partial \dot{f}_i}{\partial u_k} v_k$
    $\boldsymbol{V} = \texttt{SaveExpressions}(\boldsymbol{V}, v_i, \ddot{f}_i)$
  end if
end for

temporary variables. Therefore, the total number of intermediate and temporary variables in the output code will increase, $\geq 51n$. However, the number of operations and trigonometric functions will decrease. Consequently, the computation time for an inverse dynamics solution per cycle is further reduced.

Several Maple procedures have been developed by the authors to identify constant subexpressions and to eliminate useless expressions. In addition, the intermediate/temporary variables are recycled to reduce the total allocated memory for these variables in the output code, thus ensuring better performance.

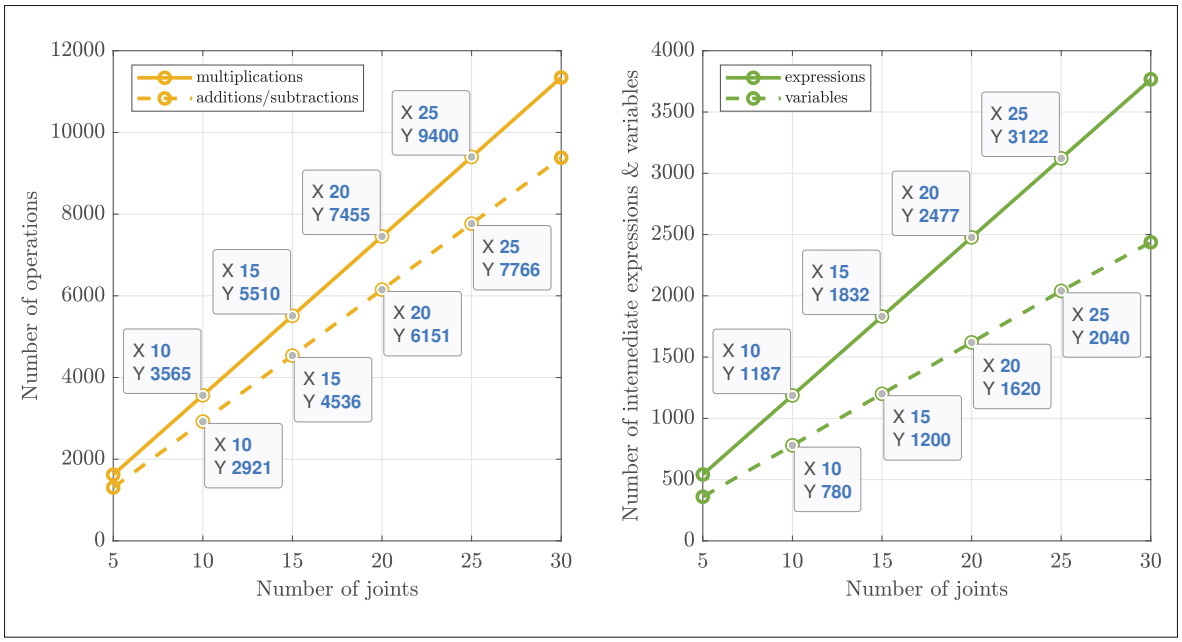

Figure 4.4 The number of operations (left), intermediate expressions/variables (right) for computing $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$

4.4.4 Computational complexity

To analyze the computational complexity of SDRNEA, six different robot models with pure revolute joints are examined with $n = (5, 10, 15, \ldots, 30)$ joints. In addition, the kinematic and dynamics parameters of these robots are chosen generally, i.e., no zero parameters in the matrices $\boldsymbol{H}$ and $\boldsymbol{\Gamma}$ (see Eq. (4.15) and Eq. (4.16)).

For each robot, symbolic expressions defined by Eq. (4.25) are generated automatically using SDRNEA implemented in Maple, version 2018. The computation time for generating optimized code varies from 82 (s) for $n = 5$ to 7047 (s) for $n = 30$, using an HP Laptop (the Intel Core i7-2720QM, CPU 2.20 GHz and 12 Gb RAM). Note that all symbolic expressions needed to evaluate $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ are generated only once using SDRNEA.

The numerical simulations for each model are conducted in Matlab in order to verify the agreement between SDRNEA and EJRNE methods. Joint motions used to compute $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$, and

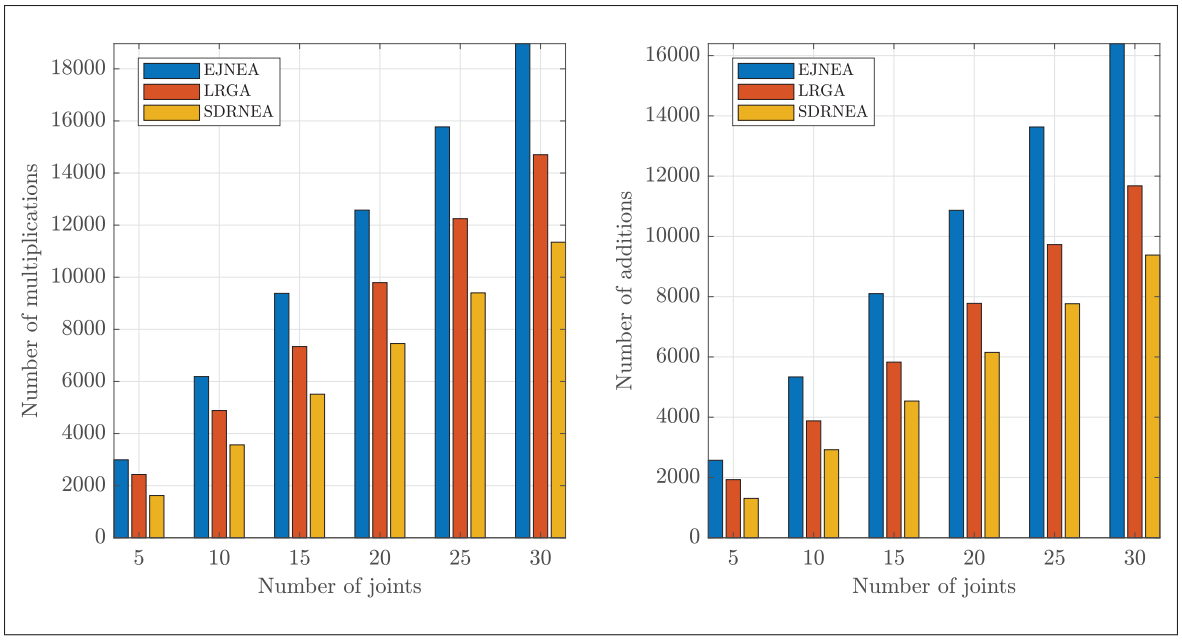

Figure 4.5 Computational complexity of three methods for computing $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$: EJNEA (Buondonno & De Luca, 2015), LRGA (Yang *et al.*, 2018), and SDRNEA

$\ddot{\boldsymbol{\tau}}$ are chosen as ninth-order polynomials (see Sect. 4.5). In addition, the kinematic and dynamic parameters of robots are chosen arbitrarily. From simulation results, the maximal derivations between two methods are approximately 1.0^{-10} (N·m) for $\boldsymbol{\tau}$ and 1.0^{-9} (N·m·s^{-1}) for $\dot{\boldsymbol{\tau}}$ and 1.0^{-8} (N·m·s^{-2}) for $\ddot{\boldsymbol{\tau}}$, due to the rounding error.

A number of operations, intermediate expressions, and variables used to evaluate $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ shown in Fig. 4.4 are measured using the codegen[cost] function in Maple. Using the polyfit function in Matlab, the computational complexity of SDRNEA is:

$$O(n) = \begin{cases} 323n - 309 & \text{additions/subtractions} \\ 389n - 325 & \text{multiplications} \end{cases} \tag{4.26}$$

Furthermore, the number of operations obtained using the SDRNEA method is compared to that obtained by other algorithms, such as EJNEA (Buondonno & De Luca, 2015) and LRGA (Yang

et al., 2018) (see page 499). Results obtained with three methods are illustrated in Fig. 4.5. It can be seen that the proposed algorithm outperforms other algorithms.

For a robot with special several kinematic and dynamic parameters, e.g., constant or zero, the number of operations obtained using the SDRNEA may be further reduced. It can be concluded that the proposed method can be used to compute the inverse dynamics of flexible-joint robots for real-time simulation and control.

4.5 Numerical simulations

In this section, the inverse dynamics of the Kuka LWR4 robot with seven flexible joints (Bischoff *et al.*, 2010) is presented to verify the effectiveness of the proposed algorithm.

4.5.1 Robot parameters

The model of the Kuka LWR4 robot shown in Fig. 4.6 has the DH parameters:

$$\boldsymbol{H} = \begin{bmatrix} q_1 & 0 & 0 & \pi/2 & 1 \\ q_2 & 0 & 0 & -\pi/2 & 1 \\ q_3 & d_3 & 0 & -\pi/2 & 1 \\ q_4 & 0 & 0 & \pi/2 & 1 \\ q_5 & d_5 & 0 & \pi/2 & 1 \\ q_6 & 0 & 0 & -\pi/2 & 1 \\ q_7 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.27}$$

and the dynamics parameters:

$$\boldsymbol{\Gamma} = \begin{bmatrix} m_1 & x_{c,1} & y_{c,1} & z_{c,1} & I_{xx,1} & I_{xy,1} & I_{xz,1} & I_{yy,1} & I_{yz,1} & I_{zz,1} \\ m_2 & x_{c,2} & y_{c,2} & z_{c,1} & I_{xx,2} & I_{xy,2} & I_{xz,2} & I_{yy,2} & I_{yz,2} & I_{zz,2} \\ m_3 & x_{c,3} & y_{c,3} & z_{c,3} & I_{xx,3} & I_{xy,3} & I_{xz,3} & I_{yy,3} & I_{yz,3} & I_{zz,3} \\ m_4 & x_{c,4} & y_{c,4} & z_{c,4} & I_{xx,4} & I_{xy,4} & I_{xz,4} & I_{yy,4} & I_{yz,4} & I_{zz,4} \\ m_5 & x_{c,5} & y_{c,5} & z_{c,5} & I_{xx,5} & I_{xy,5} & I_{xz,5} & I_{yy,5} & I_{yz,5} & I_{zz,5} \\ m_6 & x_{c,6} & y_{c,6} & z_{c,6} & I_{xx,6} & 0 & 0 & I_{yy,6} & I_{yz,6} & I_{zz,6} \\ m_7 & 0 & 0 & z_{c,7} & I_{xx,7} & 0 & 0 & I_{yy,7} & 0 & I_{zz,7} \end{bmatrix} \tag{4.28}$$

For numerical simulation, $d_3 = 0.40$ (m) and $d_5 = 0.49$ (m) while all dynamic parameters in Eq. (4.28) are referred from (Buondonno & De Luca, 2015) for a simple comparison. The spring constants are set to 1000 (N·m/rad), the friction parameters of links are set to zero. The friction matrices and rotor inertia moment of motors are given by (Jubien, Gautier & Janot, 2014):

$$\boldsymbol{D}_{v,m} = \text{diag}(14.3,\ 15.5,\ 6.50,\ 11.1,\ 4.37,\ 2.42,\ 1.75), \quad (\text{N}\cdot\text{m}\cdot\text{s/rad})$$
$$\boldsymbol{D}_{c,m} = \text{diag}(11.6,\ 10.7,\ 8.60,\ 8.02,\ 7.38,\ 4.64,\ 6.02), \quad (\text{N}\cdot\text{m}\cdot\text{s/rad})$$
$$\boldsymbol{J}_m = \text{diag}(3.20,\ 3.05,\ 1.98,\ 2.06,\ 0.801,\ 0.48,\ 0.381) \quad (\text{kg}\cdot\text{m}^2)$$

4.5.2 Trajectory planning

Joint motions of the robot are given by:

$$\boldsymbol{q}(t) = \boldsymbol{q}_0 + (\boldsymbol{q}_f - \boldsymbol{q}_0)\eta(t) \tag{4.29}$$

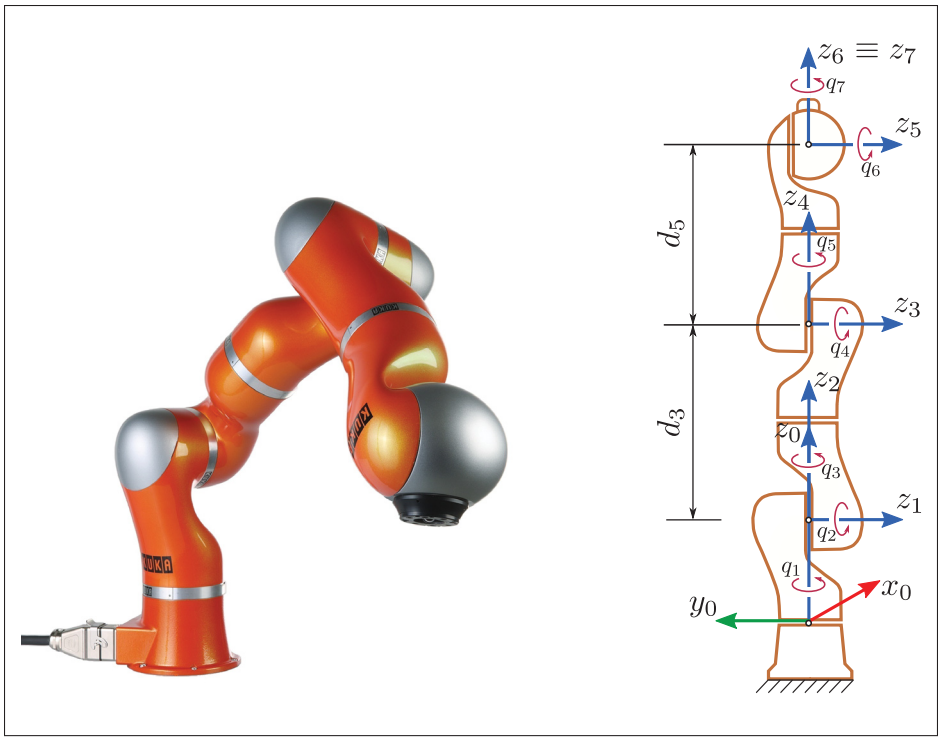

Figure 4.6 Model of the Kuka LWR4 robot (left) and the kinematic representation using DH convention (right)

where $t \in [0, T]$, T is the final time; $\boldsymbol{q}_0$ and $\boldsymbol{q}_\mathrm{f}$ are the initial and final positions. The motion profile $\eta(t)$ is defined by a ninth-order polynomial (Biagiotti & Melchiorri, 2009):

$$\eta(t) = 70\bar{t}^9 - 315\bar{t}^8 + 540\bar{t}^7 - 420\bar{t}^6 + 126\bar{t}^5 \tag{4.30}$$

where $\bar{t} = t/T$. From Eq. (4.30), the time derivatives of η including $\dot{\eta}(t)$, $\ddot{\eta}(t)$, $\dddot{\eta}(t)$, and $\ddddot{\eta}(t)$ can be computed analytically. Thus, the derivatives of $\boldsymbol{q}$ are given by:

$$\dot{\boldsymbol{q}}(t) = (\boldsymbol{q}_\mathrm{f} - \boldsymbol{q}_0)\dot{\eta}(t), \quad \ddot{\boldsymbol{q}}(t) = (\boldsymbol{q}_\mathrm{f} - \boldsymbol{q}_0)\ddot{\eta}(t), \tag{4.31}$$

$$\dddot{\boldsymbol{q}}(t) = (\boldsymbol{q}_\mathrm{f} - \boldsymbol{q}_0)\dddot{\eta}(t), \quad \ddddot{\boldsymbol{q}}(t) = (\boldsymbol{q}_\mathrm{f} - \boldsymbol{q}_0)\ddddot{\eta}(t). \tag{4.32}$$

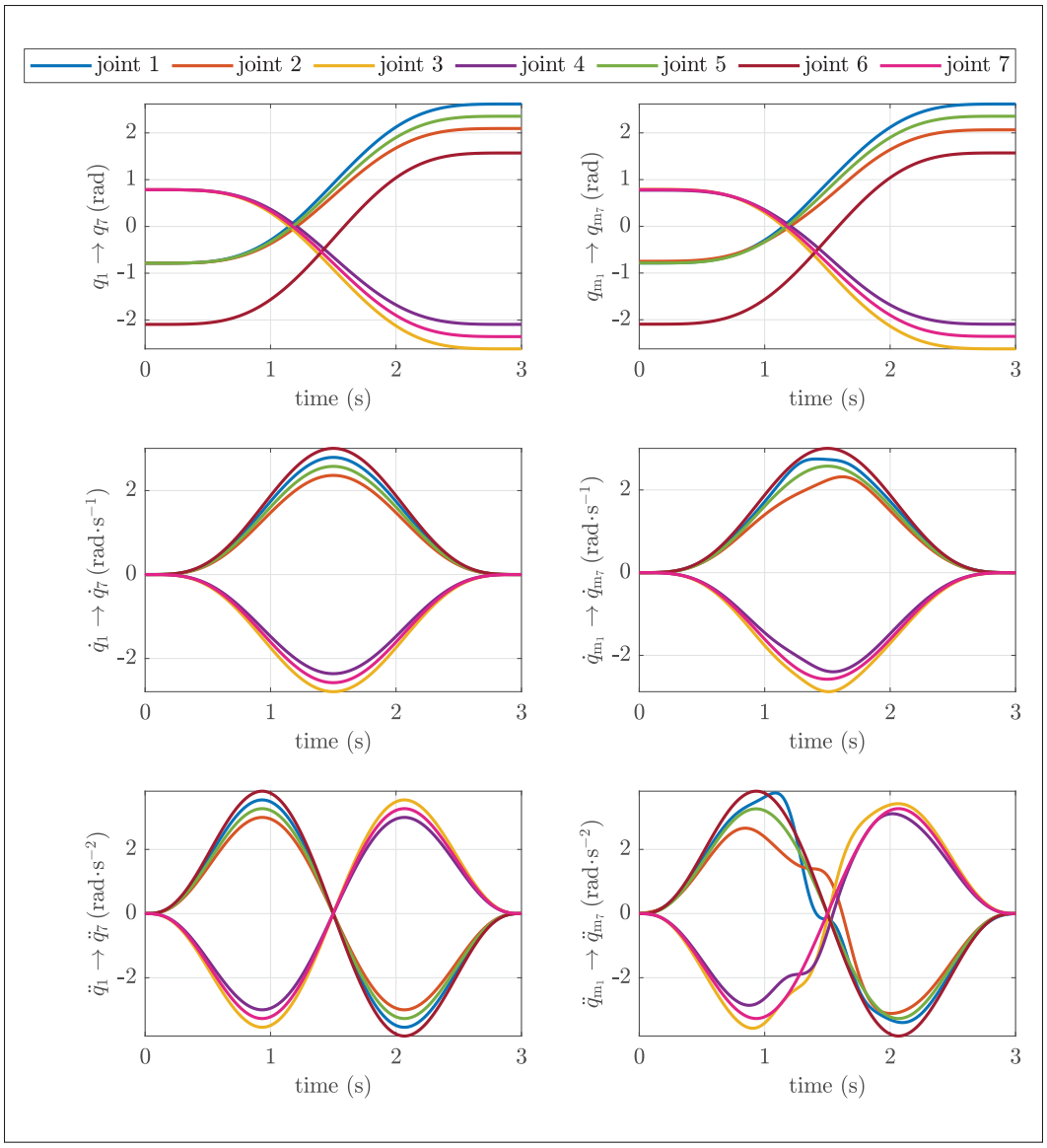

Figure 4.7 Desired positions, velocities and accelerations of joints (left) and motors (right)

4.5.3 Simulation results

The desired motions of joints and motors with $T = 3$ (s) and $\boldsymbol{q}_0^{\mathsf{T}} = \left[-\frac{\pi}{4}, -\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}, -\frac{\pi}{4}, -\frac{2\pi}{3}, \frac{\pi}{4}\right]$, $\boldsymbol{q}_{\mathrm{f}}^{\mathsf{T}} = \left[\frac{5\pi}{6}, \frac{2\pi}{3}, -\frac{5\pi}{6}, -\frac{2\pi}{3}, \frac{3\pi}{4}, \frac{\pi}{2}, -\frac{3\pi}{4}\right]$ (rad) are illustrated in Fig. 4.7. Due to the presence of

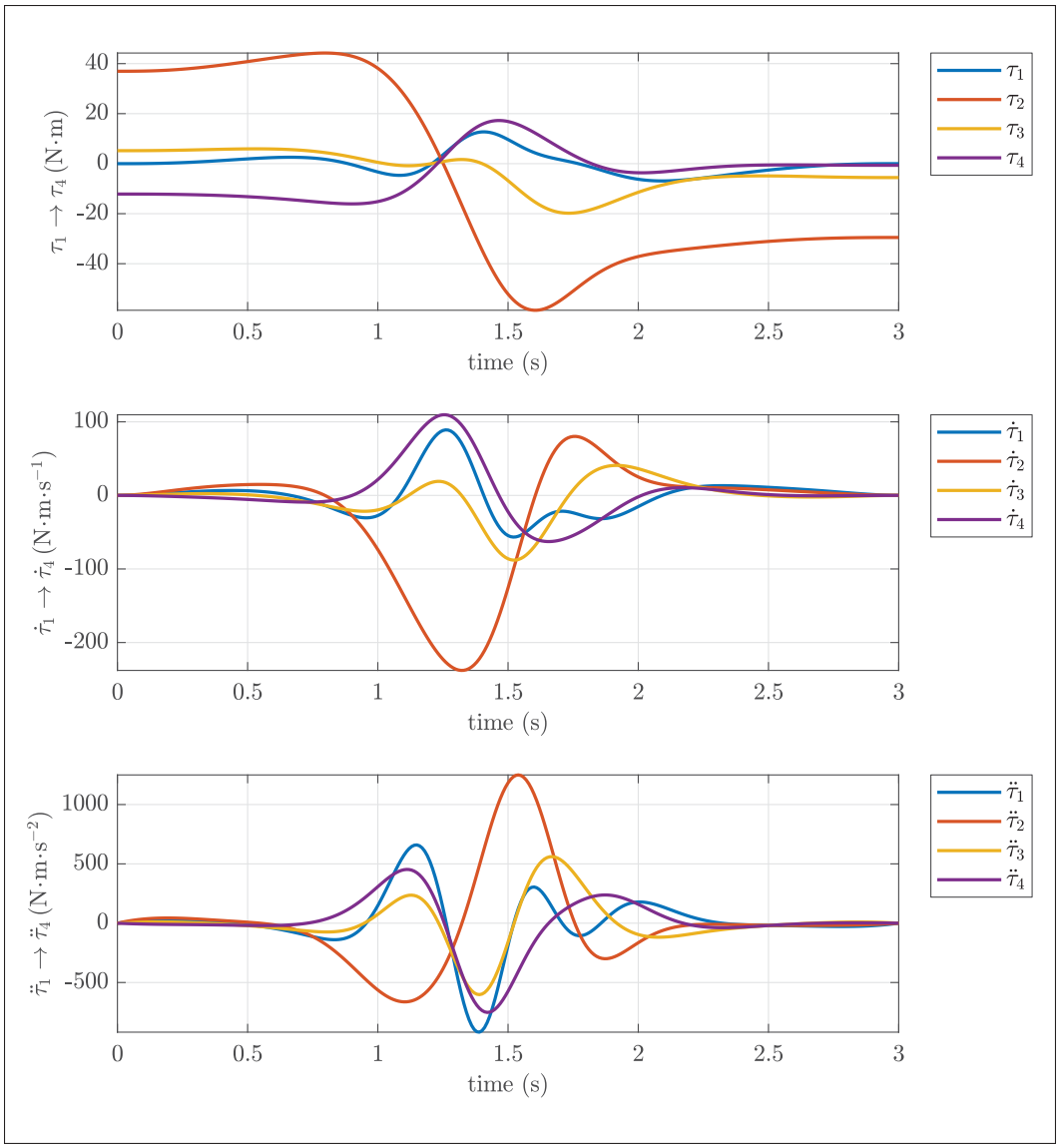

Figure 4.8 Torques and their derivatives of rigid links: joint 1 to joint 4

flexible joints, the desired motions of joints and motors are distinctive, especially for accelerations.

Numerical simulations of $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ are presented in Fig. 4.8 and Fig. 4.9. Using SDRNEA, a total of only 1289 additions/subtractions, 1516 multiplications, and 12 cos() and sin() functions are used to evaluate $\boldsymbol{\tau}$, $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$. In addition, there are 356 variables used to store 516 intermediate

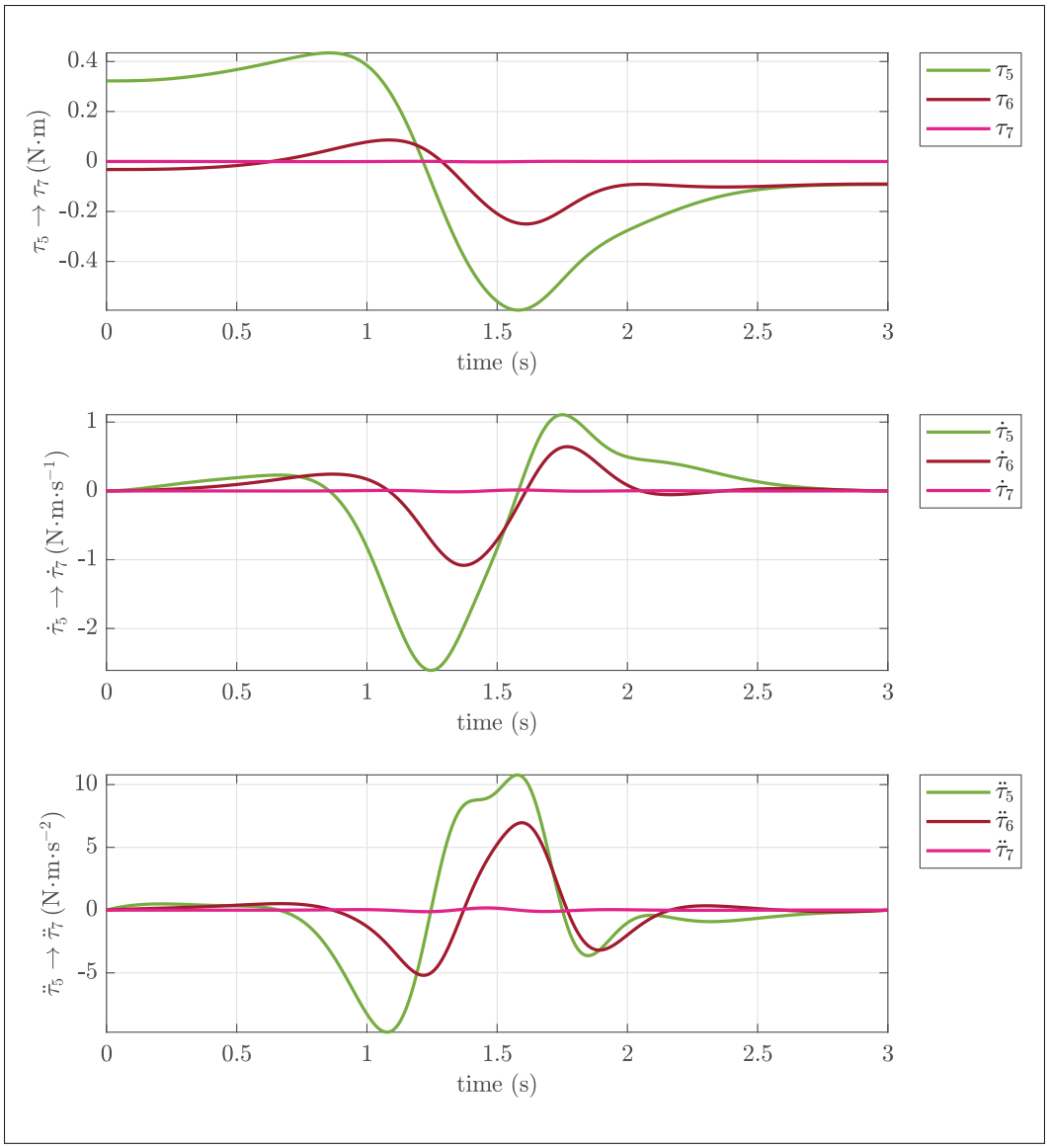

Figure 4.9 Torques and their derivatives of rigid links: joint 5 to joint 7

expressions, i.e., several variables are recycled when generating optimized code. It can be seen that the number of operations obtained for the robot is less than that presented in Eq. (4.26) because several robot parameters are zero. This means that the application of the proposed method for the Kuka LWR4 robot is even more efficient than other methods.

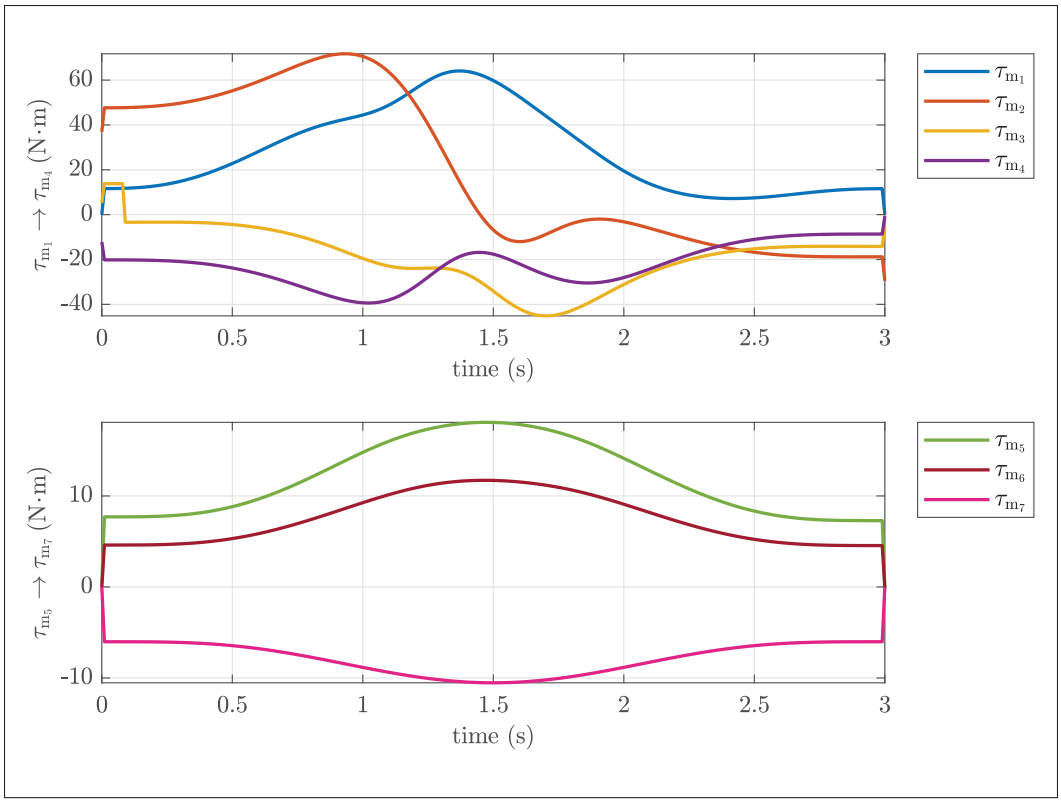

Figure 4.10 Motor torques

Finally, the desired motor torques $\boldsymbol{\tau}_{\mathrm{m}}$ are shown in Fig. 4.10. The results obtained using SDRNEA are in agreement with those obtained using EJNEA as shown in Fig. 4.11, with very small deviations, (less than 4.10^{-14} (N·m)). The computation time of $\boldsymbol{\tau}_{\mathrm{m}}$ per iteration in Matlab 2018b using SDRNEA is 0.0193 (ms), while using EJNEA is 0.0351 (ms), i.e., the proposed algorithm is run faster 1.85 times. Note that the computation time for each method is averaged by 50 thousand simulations in which $\boldsymbol{\tau}_{\mathrm{m}}$ is converted to C-code using the Coder Toolbox and mex function.

4.6 Conclusions

This paper addresses a new algorithm for automatically generating inverse dynamics of serial robots with flexible joints in symbolic form. The algorithm is developed based on the symbolic

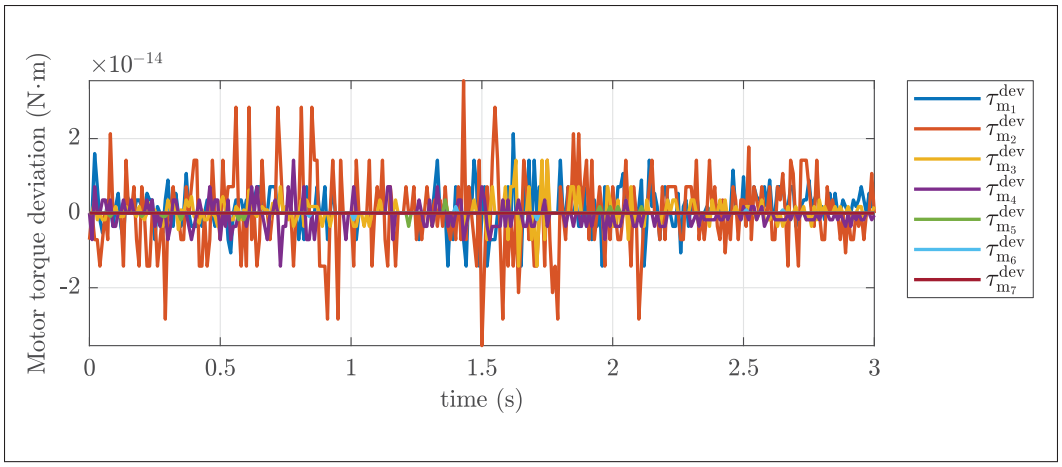

Figure 4.11 Motor driving torque deviations between EJNEA and SDRNEA methods ($=\boldsymbol{\tau}_{\mathrm{m}}^{\mathrm{EJNEA}} - \boldsymbol{\tau}_{\mathrm{m}}^{\mathrm{SDRNEA}}$)

Newton-Euler algorithm and symbolic differentiation algorithm using the computer algebra system. The inverse dynamics solution written in optimized Matlab-code or C-code can be used for model-based control. The proposed algorithm can be applied/expanded to handle the optimal trajectory planning of robots in real-time. Future work will focus on symbolic-numerical algorithms to compute the inverse dynamics of flexible-joint robots.

CONCLUSION AND RECOMMENDATIONS

Conclusion

This thesis presented dynamic models of flexible joint robots for vibration and modal analysis, as well as for parameter identification. In addition, a general algorithm was proposed to automatically generate inverse dynamics in symbolic form for robots with flexible joints, including serial elastic actuators and variable stiffness actuators.

The first paper presented a linearized model for vibration and modal analysis specifically for manipulators with flexible joints using the PD control. The proposed analytical model considers the influence of gravity and the parameters of the PD controller are taken into account. Therefore, we can compute natural frequencies and damping ratios for any desired robot configuration. To examine this model, we investigated the vibration of the Scompi robot under an impact force, in which numerical experiments simulated real-world situations (as we conducted the modal analysis). The natural frequencies and damping ratios calculated from the proposed model agree very well with those identified from the operational modal analysis technique based on the vector autoregression model.

In addition, we can evaluate/predict the robot's stability under impact forces through the natural frequencies and the damping ratios found from the proposed linearized model if the robot's parameters are predetermined accurately. Based on the similar procedures presented in this paper, we can establish new linearized models for vibration and modal analysis problems if a robot uses other controllers.

By applying the linearization model proposed in the first paper, an optimization procedure was presented in the second paper to identify the stiffness and damping parameters of flexible joint robots. These parameters play an important role in predicting the dynamic behavior of robots. We proposed constructing a new objective function based on the determinant of the characteristic

polynomial of the inverse eigenvalue problem in order to determine these parameters. The inputs of the objective function are the robot's natural frequencies and damping ratios measured at several configurations. To find unknown parameters, we proposed the use of the global optimization strategy based on a nonlinear least squares algorithm in which multiple initial values are used to find an optimal solution. Through numerical simulations for the case of a planar robot with two flexible joints, we showed that the stiffness parameters could be found with high confidence, even when the deviation of the damping ratios were up $2(\%)$. However, the damping parameters found in this way were significantly affected when the damping ratio deviation increased beyond $2(\%)$.

Finally, in the third paper, we presented a new efficient algorithm based on computer algebra systems for automatically generating the inverse dynamics of flexible joint robots. The computational complexity of our algorithm is linear with the number of a robot's joints. The proposed algorithm was developed based on the symbolic recursive Newton-Euler algorithm and symbolic differentiation technique in which all mathematical expressions (output code) obtained are written in symbolic recursive form. We proposed a framework for generating optimized output-code for real-time simulation and control purposes in which the sampling rate for control is about 1 millisecond. We showed that the inverse dynamics of the Kuka robot with six flexible joints obtained by our method and by the EJNEA method agreed with each other. Furthermore, we demonstrated that our algorithm outperformed other algorithms when comparing the numeric operations required to compute inverse dynamics. The results obtained in this paper can be applied in practical applications, especially for computing the motor torques in advanced model-based control laws, such as the feedforward control law.

Recommendations

The results obtained from this research can be further extended to improve the accuracy of the proposed mathematical models for simulation and control purposes. We recommend the following potential directions for future work.

1) In order to establish the linearized model from the closed control system for analytical modal analysis in a desired robot configuration, we need to use a control law. e.g., the PD control law with the constant gravity compensation used in the first paper. Thus, if another control law is used to control a flexible joint robot, a new linearized model must be reestablished using the Taylor expansion. To validate this model, one can compare natural frequencies and damping ratios obtained from the proposed model and those obtained through numerical simulations or experiments.

2) To investigate the vibration problem of a flexible joint robot in machining progresses, we need to establish a linearized model in which the effect of reaction forces/moments from the work-piece on the robot end-effector must be considered. Note that these forces/moments will depend on the robot's configuration. In principle, these forces/moments will change the stiffness and damping characteristics of the robot in the machining process. Therefore, we can model them as spring-damper systems with constant stiffness and damping coefficients. Using the geometric Jacobian matrix, we can combine these forces/moments into dynamic equations, and then a new linearized model can be reestablished, as in the steps presented in the first paper. If the modal information of the robot is estimated in the machining process, then we can apply the procedures presented in the second paper to identify the stiffness and damping parameters of these reaction forces/moments.

3) If a linearized model is used to identify the stiffness and damping parameters of a flexible joint, the natural frequencies and damping ratios of the robot must first be estimated at several

robot configurations. Although the software for modal analysis can be used for this purpose, they often need to measure the acceleration signals and/or exciting forces/moments using external sensors/devices. In the case of industrial robots, the motor positions can be measured using encoders that are integrated into all motors. Therefore, we need to develop efficient tools to automatically extract modal parameters from motor positions instead of acceleration signals in order to reduce the identification time. For example, the operational modal analysis technique based on the vector autoregression model can be further developed for this purpose.

4) The symbolic algorithm presented in the third paper can be applied to compute the inverse dynamics of flexible joint robots in order to design feedforward control laws. When applying this algorithm to the feedback linearization control laws (Buondonno & De Luca, 2016) in which the elastic joint torques are measured using strain gauges or extra sensors, we need to evaluate the joint accelerations and jerks online, i.e., to solve the forward dynamics problem. Therefore, this algorithm can be modified to generate automatically dynamic terms (mass matrix, Coriolis and centrifugal forces) and their time derivatives in optimized code for real-time control.

APPENDIX I

DYNAMIC MODELING AND SIMULATION

1. Derivation of dynamic equations

In this section, equations of motion of a general flexible-joint robot with prismatic and revolute joints are formulated. The effect of a robot's tool is also included in dynamic equations. All kinematic/dynamic parameters of the robot are assumed to be predetermined. Minimal coordinates of the robot include: a vector of joint coordinates $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^\mathsf{T}$ and a vector of motor coordinates after reduction $\boldsymbol{q}_\mathrm{m} = [q_{\mathrm{m},1}, q_{\mathrm{m},2}, \ldots, q_{\mathrm{m},n}]^\mathsf{T}$.

Using the standard Denavit-Hartenberg convention Denavit & Hartenberg (1955); Tsai (2003), the relative position and orientation of BF_i (body-fixed frame i at the origin O_i) with respect to BF_{i-1} (body-fixed frame $i-1$ at the origin O_{i-1}) are determined by a transformation matrix $\boldsymbol{T}_i^{(i-1)} \in \mathbb{R}^{4\times4}$:

$$\boldsymbol{T}_i^{(i-1)} = \begin{bmatrix} \boldsymbol{R}_i^{(i-1)} & \boldsymbol{p}_i^{(i-1)} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{A I-1}$$

where $\boldsymbol{p}_i^{(i-1)} \in \mathbb{R}^3$ and $\boldsymbol{R}_i^{(i-1)} \in \mathbb{R}^{3\times3}$ represent the position of O_i and orientation of BF_i, respectively. Therefore, the absolute position and orientation of the link i in IF_0 (the inertial reference frame), are derived as:

$$\boldsymbol{T}_i^{(0)} = \boldsymbol{T}_1^{(0)}\boldsymbol{T}_2^{(1)}\ldots\boldsymbol{T}_{i-1}^{(i-2)}\boldsymbol{T}_i^{(i-1)} = \begin{bmatrix} \boldsymbol{R}_i^{(0)} & \boldsymbol{p}_i^{(0)} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{A I-2}$$

When $i = n$, $\boldsymbol{T}_n^{(0)}$ represents the position and orientation of the last link.

If C_i and E are the center of mass of the link i and robot's tool, then their absolute positions in IF_0 are computed using Eq. (A I-2) as follows:

$$\boldsymbol{p}_{C_i}^{(0)} = \boldsymbol{p}_i^{(0)} + \boldsymbol{R}_i^{(0)} \boldsymbol{s}_i^{(i)} \tag{A I-3}$$

$$\boldsymbol{p}_{E}^{(0)} = \boldsymbol{p}_n^{(0)} + \boldsymbol{R}_n^{(0)} \boldsymbol{s}_E^{(n)} \tag{A I-4}$$

where $\boldsymbol{s}_i^{(i)}$ and $\boldsymbol{s}_E^{(n)} \in \mathbb{R}^3$ are the constant position vector of C_i and E in BF_i and BF_n, respectively. To take time derivative of Eq. (A I-3) and Eq. (A I-4), one obtains $\dot{\boldsymbol{p}}_{C_i}^{(0)}$ and $\dot{\boldsymbol{p}}_{E}^{(0)}$.

The angular velocity matrix of link i in BF_i is defined by Siciliano *et al.* (2009):

$$\widehat{\boldsymbol{\omega}}_i^{(i)} = (\boldsymbol{R}_i^{(0)})^{\mathsf{T}} (\dot{\boldsymbol{R}}_i^{(0)}) = \begin{bmatrix} 0 & -\omega_{i,z}^{(i)} & \omega_{i,y}^{(i)} \\ \omega_{i,z}^{(i)} & 0 & -\omega_{i,x}^{(i)} \\ -\omega_{i,y}^{(i)} & \omega_{i,x}^{(i)} & 0 \end{bmatrix} \tag{A I-5}$$

and its angular velocity vector in BF_i is:

$$\boldsymbol{\omega}_i^{(i)} = \begin{bmatrix} \omega_{i,x}^{(i)} & \omega_{i,y}^{(i)} & \omega_{i,z}^{(i)} \end{bmatrix}^{\mathsf{T}} \tag{A I-6}$$

The angular velocity matrix of the robot's tool in IF_0 is computed as Siciliano *et al.* (2009):

$$\widehat{\boldsymbol{\omega}}_E^{(0)} = (\dot{\boldsymbol{R}}_n^{(0)})(\boldsymbol{R}_n^{(0)})^{\mathsf{T}} = \begin{bmatrix} 0 & -\omega_{n,z}^{(0)} & \omega_{n,y}^{(0)} \\ \omega_{n,z}^{(0)} & 0 & -\omega_{n,x}^{(0)} \\ -\omega_{n,y}^{(0)} & \omega_{n,x}^{(0)} & 0 \end{bmatrix} \tag{A I-7}$$

and its angular velocity vector in IF_0 is:

$$\boldsymbol{\omega}_E^{(0)} = \begin{bmatrix} \omega_{n,x}^{(0)} & \omega_{n,y}^{(i)} & \omega_{n,z}^{(0)} \end{bmatrix}^{\mathsf{T}} \tag{A I-8}$$

Using the assumptions mentioned in Section 2.2.1, the angular velocity vector of the rotor i in RF_i (rotor-fixed frame i) is reduced to De Luca & Book (2016):

$$\boldsymbol{\omega}_{\mathrm{r}_i}^{(\mathrm{r}_i)} = \boldsymbol{R}_{i-1}^{(\mathrm{r}_i)} \boldsymbol{\omega}_{i-1}^{(i-1)} + \begin{bmatrix} 0 & 0 & N_i \dot{q}_{\mathrm{m},i} \end{bmatrix}^{\mathsf{T}} \approx \begin{bmatrix} 0 & 0 & N_i \dot{q}_{\mathrm{m},i} \end{bmatrix}^{\mathsf{T}} \tag{A I-9}$$

where $\boldsymbol{R}_{i-1}^{(\mathrm{r}_i)}$ is the constant rotation matrix of BF_{i-1} with respect to RF_i, N_i (> 100) and $q_{\mathrm{m},i}$ is the gear ratio and coordinate of the motor i, respectively. The linear velocity of the rotor's center of mass does not need to be computed because the mass of motor i is combined into the mass of link $i-1$.

The torsional deflection e_i of flexible joint i is defined as follows:

$$e_i = \begin{cases} q_i - q_{\mathrm{m},i} & \text{if the joint } i \text{ is revolute} \\ \\ \frac{q_i}{r_i} - q_{\mathrm{m},i} & \text{if the joint } i \text{ is prismatic} \end{cases} \tag{A I-10}$$

where r_i represents the 'characteristic radius' of the joint's driver i which is used to convert the rotational motion of the motor i to the translational motion of the joint i.

The kinetic energy of all links, the robot's tool and rotors is computed as follows:

$$\begin{aligned} K =& \frac{1}{2} \sum_{i=1}^{n} \left(m_i (\dot{\boldsymbol{p}}_{\mathrm{C}_i}^{(0)})^{\mathsf{T}} \dot{\boldsymbol{p}}_{\mathrm{C}_i}^{(0)} + (\boldsymbol{\omega}_i^{(i)})^{\mathsf{T}} \boldsymbol{I}_i^{(i)} \boldsymbol{\omega}_i^{(i)} \right) + \\ & \frac{1}{2} m_{\mathrm{E}} (\dot{\boldsymbol{p}}_{\mathrm{E}}^{(0)})^{\mathsf{T}} \dot{\boldsymbol{p}}_{\mathrm{E}}^{(0)} + \frac{1}{2} (\boldsymbol{\omega}_{\mathrm{E}}^{(0)})^{\mathsf{T}} \boldsymbol{I}_{\mathrm{E}}^{(0)} \boldsymbol{\omega}_{\mathrm{E}}^{(0)} + \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{\omega}_{\mathrm{r}_i}^{(\mathrm{r}_i)})^{\mathsf{T}} \boldsymbol{I}_{\mathrm{r}_i}^{(\mathrm{r}_i)} \boldsymbol{\omega}_{\mathrm{r}_i}^{(\mathrm{r}_i)} \\ =& \frac{1}{2} \dot{\boldsymbol{q}}^{\mathsf{T}} \boldsymbol{M} \dot{\boldsymbol{q}} + \frac{1}{2} \dot{\boldsymbol{q}}_{\mathrm{m}}^{\mathsf{T}} \boldsymbol{B} \dot{\boldsymbol{q}}_{\mathrm{m}} \end{aligned} \tag{A I-11}$$

where m_i and $\boldsymbol{I}_i^{(i)} \in \mathbb{R}^{3\times3}$ are the mass and inertia tensor of the link i about C_i in BF_i; m_{E} and $\boldsymbol{I}_{\mathrm{E}}^{(0)}$ is the mass and inertia tensor of the robot's tool about E in IF_0; $\boldsymbol{I}_{\mathrm{r}_i}^{(\mathrm{r}_i)} \in \mathbb{R}^{3\times3}$ is the inertia tensor of rotor i in RF_i; $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ are the mass matrix of the robot's links and tool. The matrix $\boldsymbol{B}$ is defined by:

$$\boldsymbol{B} = \mathrm{diag}(N_1^2 I_{\mathrm{r}_1,z}, N_2^2 I_{\mathrm{r}_2,z}, \ldots, N_n^2 I_{\mathrm{r}_n,z}) \tag{A I-12}$$

where $I_{\mathrm{r}_i,z}$ is the inertial moment of the rotor i about its rotating z-axis.

The potential energy is computed as follows:

$$\begin{aligned} P &= \frac{1}{2}\sum_{i=1}^{n} k_i e_i^2 - \sum_{i=1}^{n} m_i \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_{\mathrm{C}_i}^{(0)} - m_{\mathrm{E}} \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_{\mathrm{E}}^{(0)} \\ &= \frac{1}{2}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}})^{\mathsf{T}} \boldsymbol{K}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\mathrm{m}}) - \sum_{i=1}^{n} m_i \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_{\mathrm{C}_i}^{(0)} - m_{\mathrm{E}} \boldsymbol{g}_0^{\mathsf{T}} \boldsymbol{p}_{\mathrm{E}}^{(0)} \end{aligned} \tag{A I-13}$$

where k_i the stiffness coefficient of the motor shaft i; $\boldsymbol{g}_0 \in \mathbb{R}^3$ is the gravity vector in IF_0. Here, the matrices $\boldsymbol{W}$ and $\boldsymbol{K}$ are defined by:

$$\boldsymbol{W} = \mathrm{diag}(\frac{1}{r_1}, \frac{1}{r_2}, \ldots, \frac{1}{r_n}) \tag{A I-14}$$

$$\boldsymbol{K} = \mathrm{diag}(k_1, k_2, \ldots, k_n), \tag{A I-15}$$

Note that $r_i = 1$ if the joint i is revolute, $i = (1, \ldots, n)$.

The Rayleigh dissipation function is computed as follows:

$$\Phi = \frac{1}{2}\sum_{i=1}^{n} d_{\mathrm{f},i}\dot{q}_i^2 + \frac{1}{2}\sum_{i=1}^{n} d_{\mathrm{fm},i}\dot{q}_{\mathrm{m},i}^2 = \frac{1}{2}\dot{\boldsymbol{q}}^{\mathsf{T}}\boldsymbol{D}\dot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}_{\mathrm{m}}^{\mathsf{T}}\boldsymbol{D}_{\mathrm{m}}\dot{\boldsymbol{q}}_{\mathrm{m}} \tag{A I-16}$$

where $d_{\mathrm{f},i}$ and $d_{\mathrm{fm},i}$ are viscous coefficients of joint i on the link side and on the motor side, respectively.

$$\boldsymbol{D} = \mathrm{diag}(d_{\mathrm{f},1}, d_{\mathrm{f},2}, \ldots, d_{\mathrm{f},n}), \tag{A I-17}$$

$$\boldsymbol{D}_{\mathrm{m}} = \mathrm{diag}(d_{\mathrm{fm},1}, d_{\mathrm{fm},2}, \ldots, d_{\mathrm{fm},n}) \tag{A I-18}$$

The virtual work of nonconservative forces is computed as follows:

$$\delta W = (\boldsymbol{J}^{\mathsf{T}}\boldsymbol{\tau}_{\mathrm{ext}}^{(0)})^{\mathsf{T}}\delta\boldsymbol{q} + \boldsymbol{\tau}_{\mathrm{m}}^{\mathsf{T}}\delta\boldsymbol{q}_{\mathrm{m}} \tag{A I-19}$$

where $\boldsymbol{\tau}_{\text{ext}}^{(0)} \in \mathbb{R}^6$ is the vector of external forces and moments applied at E in IF_0; $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the geometric Jacobian matrix Siciliano *et al.* (2009) derived from $\dot{\boldsymbol{p}}_{\text{E}}^{(0)}$ and $\boldsymbol{\omega}_{\text{E}}^{(0)}$; $\boldsymbol{\tau}_{\text{m}} \in \mathbb{R}^n$ is the vector of motor torques; $\delta\boldsymbol{q}$ and $\delta\boldsymbol{q}_{\text{m}}$ are vectors of virtual displacements.

From all kinematic/dynamic quantities presented above, equations of motion of the flexible-joint robot with n links and the robot's tool using the Lagrangian formulation Siciliano *et al.* (2009) are expressed as follows:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{W}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\text{m}}) = \boldsymbol{J}^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{\tau}_{\text{ext}} \tag{A I-20}$$

$$\boldsymbol{B}\ddot{\boldsymbol{q}}_{\text{m}} + \boldsymbol{D}_{\text{m}}\dot{\boldsymbol{q}}_{\text{m}} - \boldsymbol{K}(\boldsymbol{W}\boldsymbol{q} - \boldsymbol{q}_{\text{m}}) = \boldsymbol{\tau}_{\text{m}} \tag{A I-21}$$

where $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is the matrix of centrifugal/Coriolis terms and $\boldsymbol{g} \in \mathbb{R}^n$ is the vector of gravity terms.

2. Dynamic simulation for modal analysis

This section presents simulation results of the Scompi robot presented in Section 2.4.3 in which the mass $m_{\text{E}} = 4$ (kg) is added to the end-effector.

The motor torques are computed using the control law presented in Eq. (2.2) and the results obtained are shown in Fig. I-1. In addition, the linear accelerations of the end-effector $\ddot{\boldsymbol{x}}_{\text{E}}$ are derived from Eq. (A I-4) in which $\ddot{\boldsymbol{x}}_{\text{E}}$ depends on joint variables, such as $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. Based on Eq. (2.1), these variables are computed using numerical integration methods. The simulation results of $\ddot{\boldsymbol{x}}_{\text{E}}$ are plotted in Fig. I-2. Using NAFID-tool, eight stable poles are identified as shown in Fig. I-3.

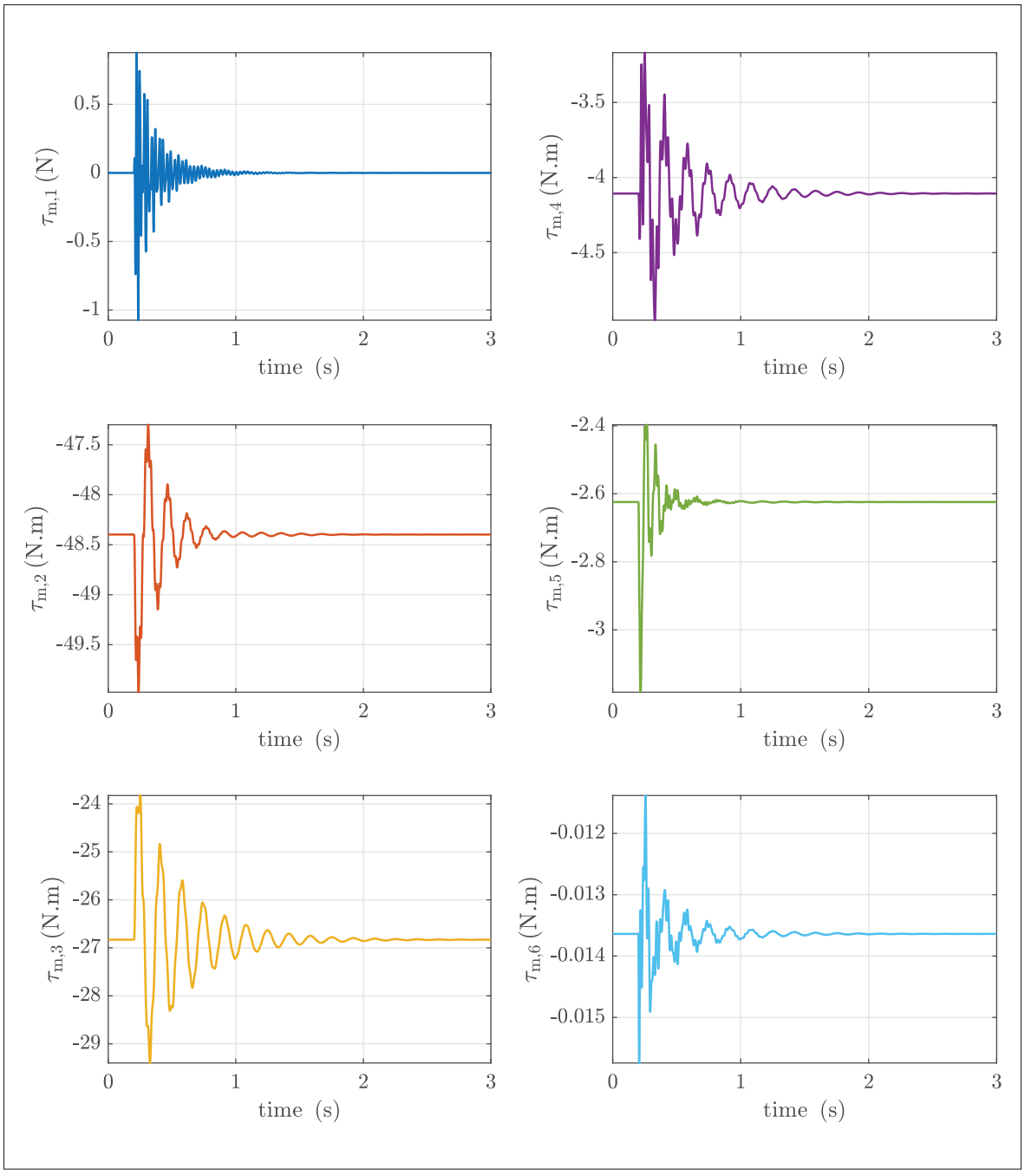

Figure-A I-1 Motor torques (case 2)

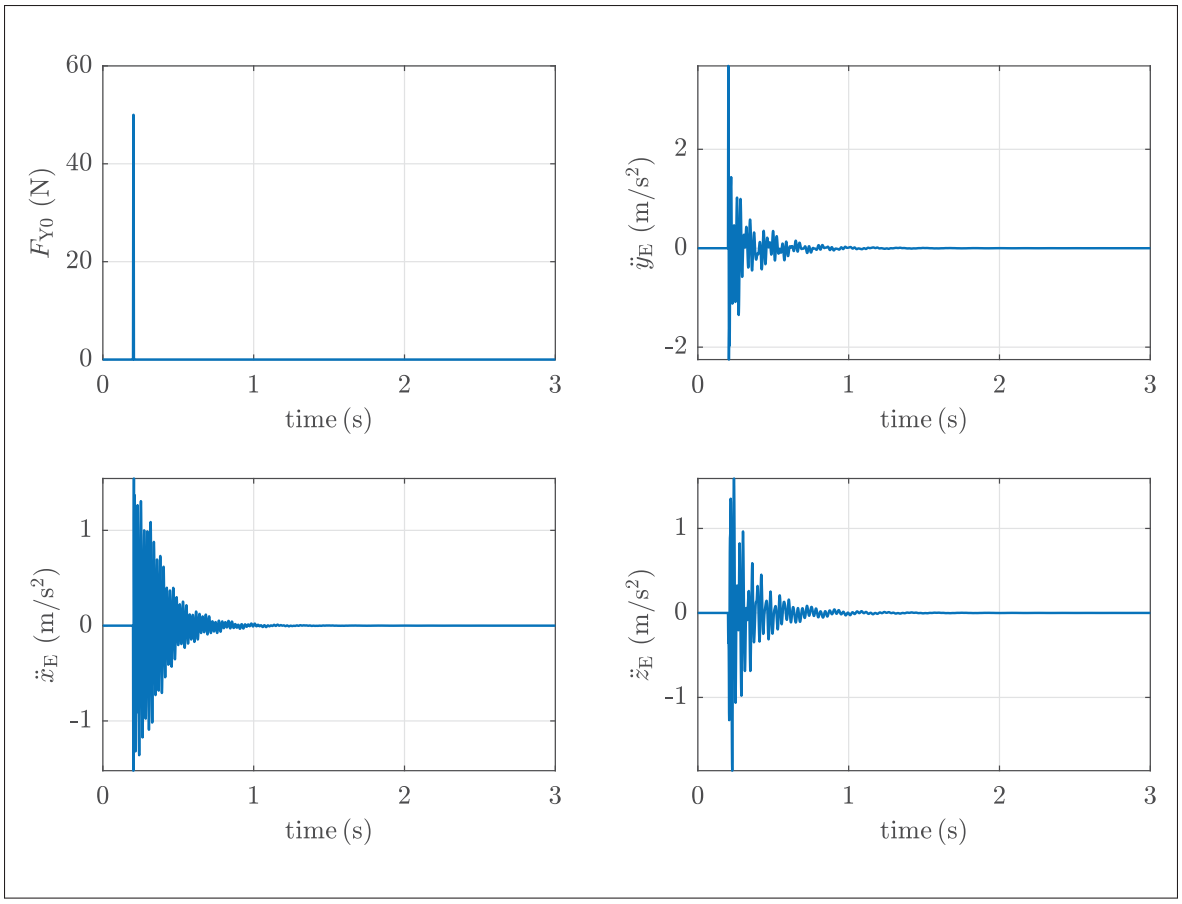

Figure-A I-2 Impact force and linear accelerations of the end-effector (case 2)

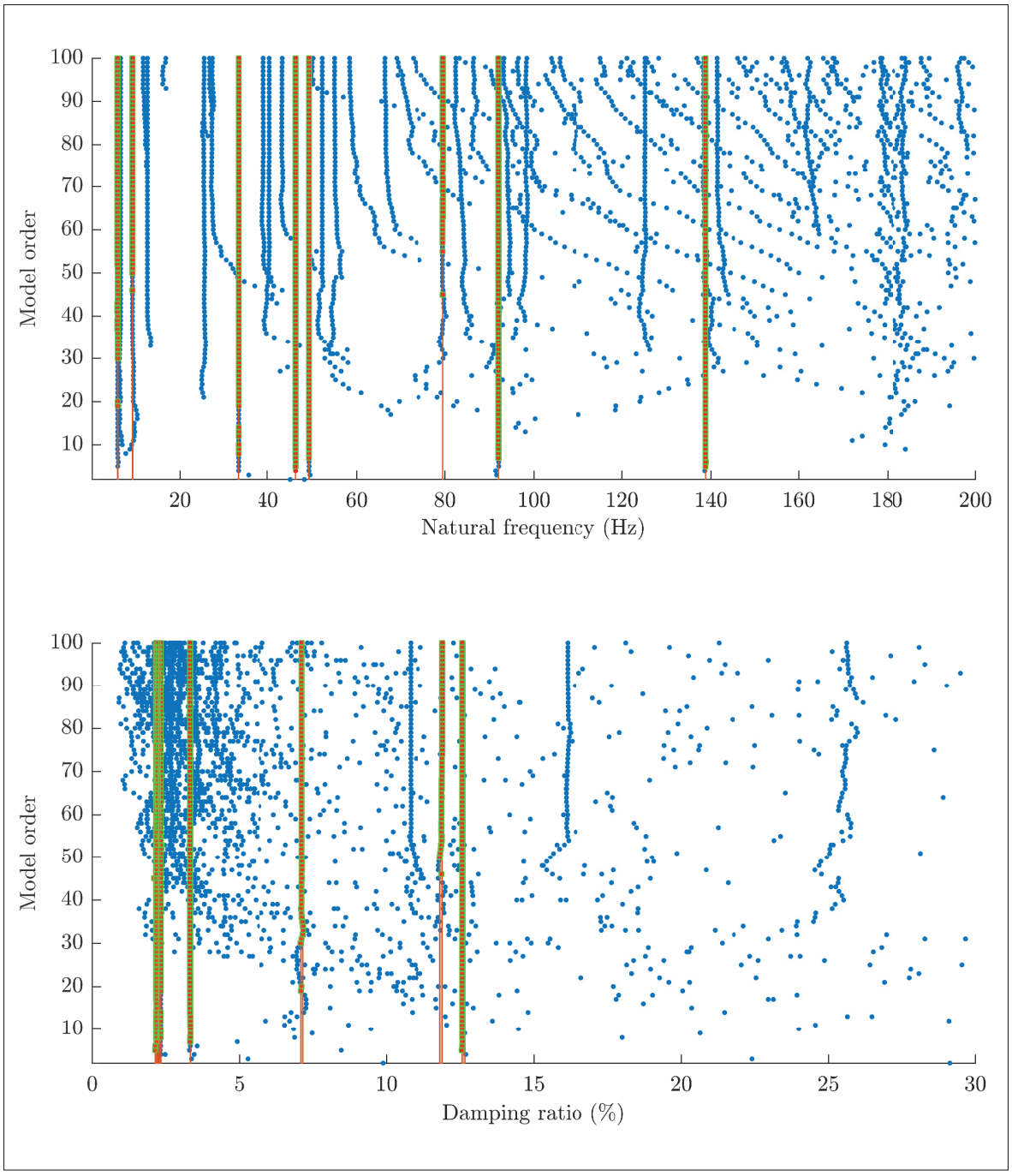

Figure-A I-3 Stable natural frequencies and damping ratios (green color) are identified from the end-effector accelerations (case 2)

APPENDIX II

COMPUTATION OF MODAL PARAMETERS

We assume that the joint coordinates q_1 and q_2 are in the intervals: $0 \leq q_1 \leq 2\pi$ and $0 \leq q_2 \leq \pi$, the natural frequencies and damping ratios of the robot are computed as follows:

- Step 1: Compute $q_{1,i} = i\frac{\pi}{p}$ and $q_{2,j} = j\frac{\pi}{p}$ for $i = 0, \ldots, 2p$ and $j = 0, \ldots, p$.
- Step 2: Evaluate the linearized matrices $\overline{\boldsymbol{M}}$, $\overline{\boldsymbol{D}}$, and $\overline{\boldsymbol{K}}$ for each configuration $(q_{1,i}, q_{2,j})$.
- Step 3: Compute the matrix square root $\overline{\boldsymbol{S}}$ from the matrix $\overline{\boldsymbol{M}}$.
- Step 4: Solve for the eigenvalues using Eq. (3.17), and the natural frequencies damping ratios using Eq. (3.18) and Eq. (3.19), respectively.

Note that the matrix square root in Step 3 can be obtained using the sqrtm function, while the eigenvalues in Step 4 can be computed using the polyeig function in Matlab.

For example, when $p = 720$, the natural frequencies and damping ratios in the joint space are shown from Fig. II-1 to Fig. II-4 in which the blue points with indices are used to identify the stiffness and damping parameters.

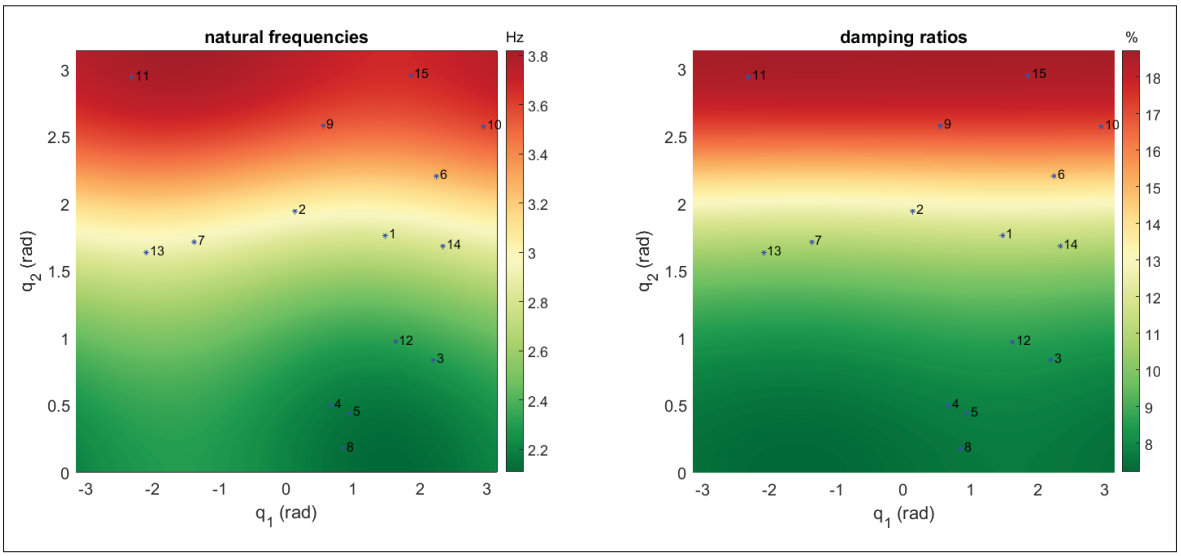

Figure-A II-1 Natural frequencies and damping ratios of the first mode in the joint space

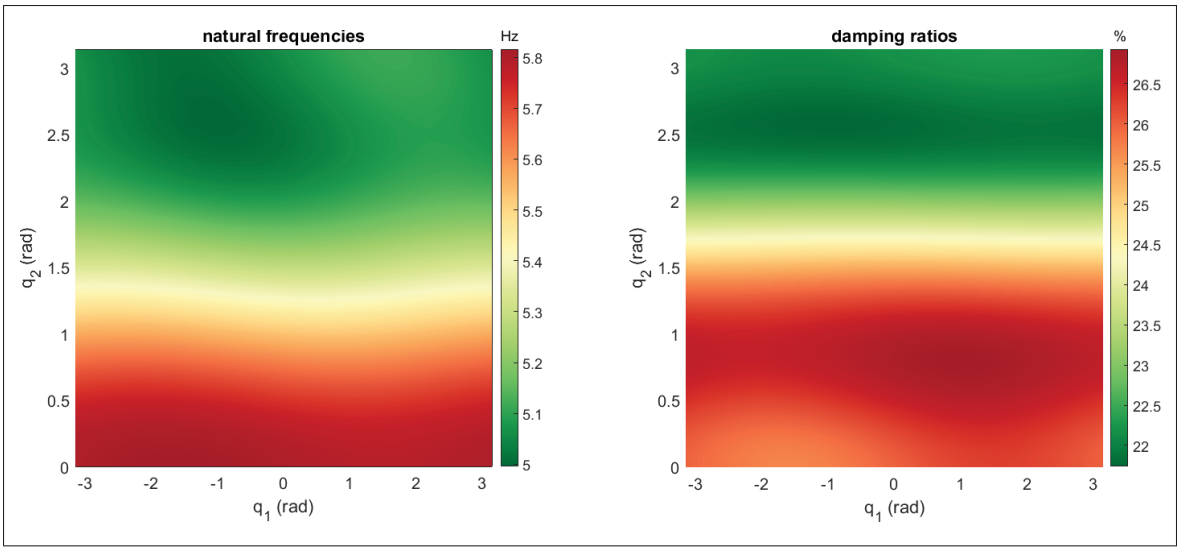

Figure-A II-2 Natural frequencies and damping ratios of the second mode in the joint space

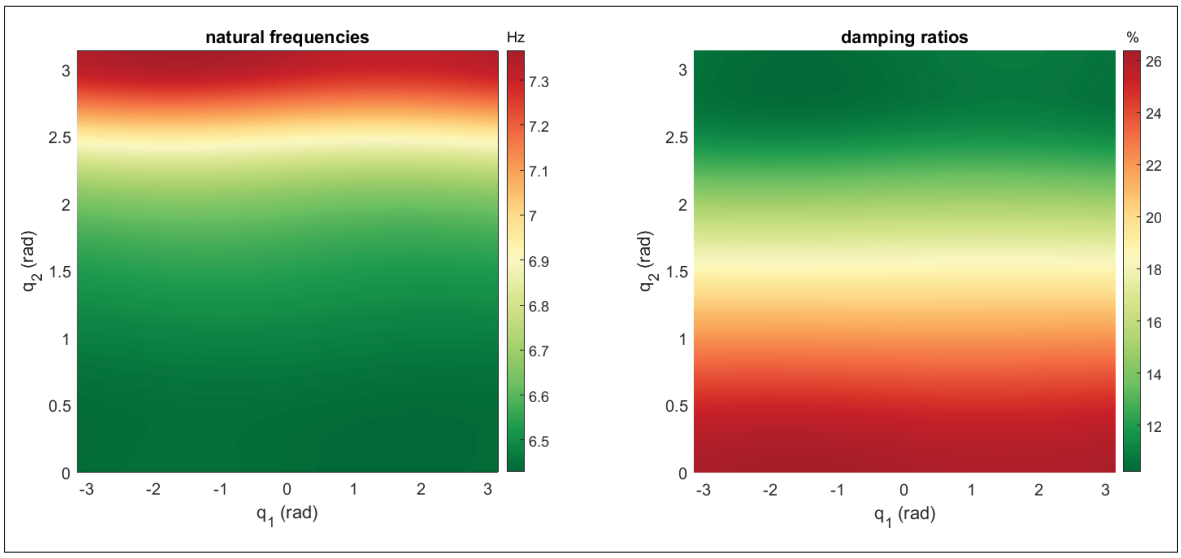

Figure-A II-3 Natural frequencies and damping ratios of the third mode in the joint space

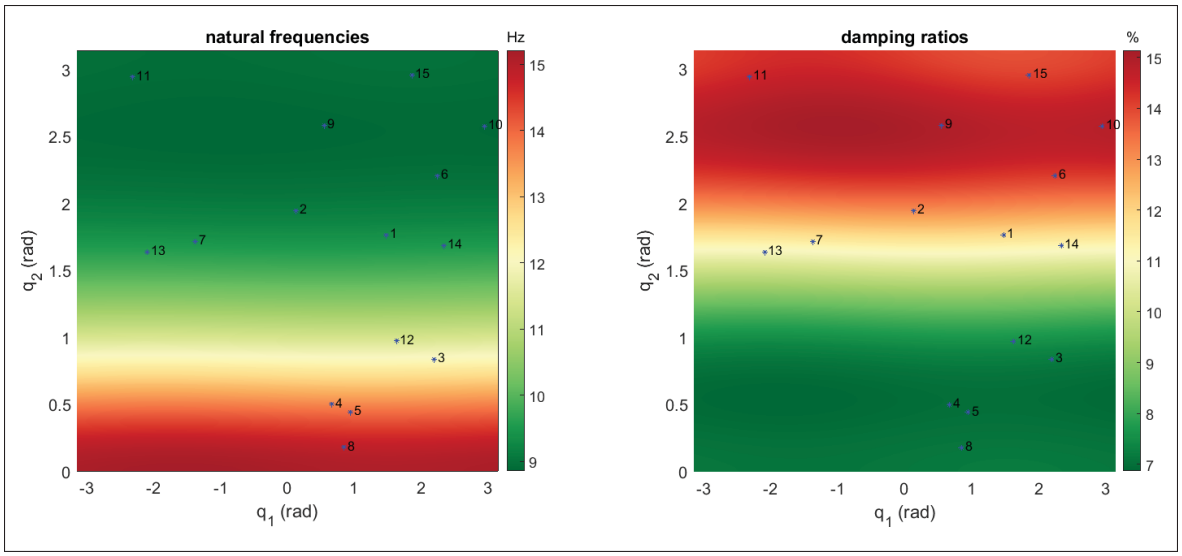

Figure-A II-4 Natural frequencies and damping ratios of the fourth mode in the joint space

BIBLIOGRAPHY

Albu-Schäffer, A., Ott, C. & Hirzinger, G. (2007). A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Rob. Res.*, 26(1), 23-39. doi: 10.1177/0278364907073776.

Angeles, J. (2014). *Fundamentals of Robotic Mechanical Systems Theory, Methods, and Algorithms* (ed. 4th). Springer.

Armstrong-Helouvry, B. (1991). *Control of Machines with Friction*. Kluwer Academic Publishers.

Berninger, T. F. C., Fuderer, S. & Rixen, D. J. (2020). Modal Analysis of a 7 DoF Sweet Pepper Harvesting Robot. *Topics in Modal Analysis and Testing*, pp. 163-170.

Bernzen, W. (1999). Active Vibration Control of Flexible Robots Using Virtual Spring-damper Systems. *Journal of Intelligent and Robotic Systems*, (24), 69-88. doi: 10.1023/A:1008035116904.

Biagiotti, L. & Melchiorri, C. (2009). *Trajectory Planning for Automatic Machines and Robots*. Springer.

Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G. & Hirzinger, G. (2010). The Kuka-DLR Lightweight Robot arm-a new reference platform for robotics research and manufacturing. *41st International Symposium on Robotics*, pp. 1-8.

Bisu, C.-F., Cherif, M., Gérard, A. & KNevez, J.-Y. (2012). Dynamic behavior analysis for a six axis industrial machining robot. *Journal of Advanced Materials Research*, 423, 65-76.

Bottin, M., Cocuzza, S., Comand, N. & Doria, A. (2020). Modeling and Identification of an Industrial Robot with a Selective Modal Approach. *Applied Sciences*, 10(13). doi: 10.3390/app10134619.

Brincker, R. & Ventura, C. (2015). *Introduction to Operational Modal Analysis*. Wiley. doi: 10.1002/9781118535141.

Buondonno, G. & De Luca, A. (2015). A recursive Newton-Euler algorithm for robots with elastic joints and its application to control. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5526-5532. doi: 10.1109/iros.2015.7354160.

Buondonno, G. & De Luca, A. (2016). Efficient computation of inverse dynamics and feedback linearization for VSA-based robots. *IEEE Robot. Autom. Lett.*, 1(2), 908-915. doi: 10.1109/lra.2016.2526072.

Cara, F. J., Juan, J., Alarcon, E., Reynders, E. & De Roeck, G. (2013). Modal contribution and state space order selection in operational modal analysis. *Mechanical Systems and Signal Processing*, 38(2), 276-298. doi: 10.1016/j.ymssp.2013.03.001.

Chu, M., Zhang, Y., Chen, G. & Sun, H. (2015). Effects of joint controller on analytical modal analysis of rotational flexible manipulator. *Chinese Journal of Mechanical Engineering*, 3(28), 460-469. doi: 10.3901/CJME.2015.0121.017.

Chu, M.-T. & Golub, G.-H. (2005). *Inverse eigenvalue problems: theory, algorithms, and applications*. Oxford University Press.

Chung, W., Fu, L. & Kröger, T. (2016). Motion control. In *Springer Handbook of Robotics* (pp. 163-194). Springer. doi: 10.1007/978-3-319-32552-1_8.

De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 233-240. doi: 10.1109/ROBOT.2000.844064.

De Luca, A. & Book, W. (2016). Robots with Flexible Elements. In Siciliano, B. & Khatib, O. (Eds.), *Springer Handbook of Robotics* (ed. 2, pp. 243-282). Springer. doi: 10.1007/978-3-319-32552-1_11.

De Luca, A. & Tomei, P. (1996). Elastic joints. In Canudas de Wit, C., Siciliano, B. & Bastin, G. (Eds.), *Theory of Robot Control* (pp. 179-217). Springer-Verlag.

De Luca, A., Farina, R. & Lucibello, P. (2005). On the Control of Robots with Visco-Elastic Joints. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4297-4302. doi: 10.1109/ROBOT.2005.1570781.

De Luca, A. (1998). Decoupling and feedback linearization of robots with mixed rigid/elastic joints. *International Journal of Robust and Nonlinear Control*, 8(11), 965-977. doi: 10.1002/(SICI)1099-1239.

Denavit, J. & Hartenberg, R. (1955). A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *J. Appl. Mech.*

Devriendt, C. & Guillaume, P. (2008). Identification of modal parameters from transmissibility measurements. *J. Sound Vib.*, 314(1), 343-356. doi: doi.org/10.1016/j.jsv.2007.12.022.

Do, T.-T., Vu, V.-H. & Liu, Z. (2019). NAFID-A grid tool for output only modal analysis. *Proceedings SURVISHNO*, pp. 119-127.

Do, T.-T., Vu, V.-H. & Liu, Z. (2022). Linearization of dynamic equations for vibration and modal analysis of flexible joint manipulators. *Mech Mach Theory*, 167. doi: 10.1115/1.4051355.

Dumas, C., Caro, S., Garnier, S. & Furet, B. (2011). Joint stiffness identification of six-revolute industrial serial robots. *Robotics and Computer-Integrated Manufacturing*, 27(4), 881-888. doi: 10.1016/j.rcim.2011.02.003.

Dwivedy, S. & Eberhard, P. (2006). Dynamic analysis of flexible manipulators, a literature review. *Mech Mach Theory*, 41(7), 749-777.

Ewins, D. J. (2000). *Modal testing: Theory, Practical and Applications* (ed. second). Research Studies Press.

Featherstone, R. (2008). *Rigid Bodies Dynamics Algorithms*. Springer.

Friswell, M. I. & Mottershead, J. E. (1995). *Finite Element Model Updating in Structural Dynamics*. Kluwer Academic Publishers.

Gautier, M., Jubien, A., Janot, A. & Robet, P.-P. (2013). Dynamic Identification of flexible joint manipulators with an efficient closed loop output error method based on motor torque output data. *2013 IEEE International Conference on Robotics and Automation*, pp. 2949-2955. doi: 10.1109/ICRA.2013.6630986.

Ginsberg, J. H. (1998). *Advanced Engineering Dynamics* (ed. Second). Cambridge University Press.

Giusti, A., Malzahn, J., Tsagarakis, N. & Althoff, M. (2018). On the combined inverse-dynamics/passivity-based control of elastic-joint robots. *IEEE Trans. Robot.*, 34(6), 1461-1471. doi: 10.1109/tro.2018.2861917.

Guarino Lo Bianco, C. (2009). Evaluation of generalized force derivatives by means of a recursive Newton-Euler approach. *IEEE Trans. Robot.*, 25(4), 954-959. doi: 10.1109/tro.2009.2024787.

Hao, D., Wang, W., Liu, Z. & Yun, C. (2020). Experimental study of stability prediction for high-speed robotic milling of aluminum. *Journal of Vibration and Control*, 26(7-8), 387-398. doi: 10.1177/1077546319880376.

Hazel, B., Côté, J., Laroche, Y. & Mongenot, P. (2012). A portable, multiprocess, track-based robot for in situ work on hydropower equipment. *J. Field Rob.*, 21(1), 69-101. doi: 10.1002/rob.20425.

Hollerbach, J., Khalil, W. & Gautier, M. (2016). Model Identification. In Siciliano, B. & Khatib, O. (Eds.), *Springer Handbook of Robotics* (ed. 2, pp. 113-138). Springer. doi: 10.1007/978-3-319-32552-1_6.

Hollerbach, J. (1980). A recursive Lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Syst., Man, Cybern.*, 10(11), 730-736. doi: 10.1109/tsmc.1980.4308393.

Huynh, H., Assadi, H., Dambly, V., Rivière-Lorphèvre, E. & Verlinden, O. (2021). Direct method for updating flexible multibody systems applied to a milling robot. *Robotics and Computer-Integrated Manufacturing*, 68. doi: 10.1016/j.rcim.2020.102049.

Iglesias, I., Sebastián, M. & Ares, J. (2015). Overview of the State of Robotic Machining: Current Situation and Future Potential. *Procedia Engineering*, 132, 911-917. doi: 10.1016/j.proeng.2015.12.577.

Inman, D. J. (2017). *Vibration with Control* (ed. Second). John Wiley & Sons.

Jubien, A., Gautier, M. & Janot, A. (2014). Dynamic identification of the Kuka LWR robot using motor torques and joint torque sensors data. *19th IFAC World Congress*, pp. 8391-8396. doi: 10.3182/20140824-6-za-1003.01079.

Kane, T. & Levinson, D. (1983). The use of Kane's dynamical equations in robotics. *Int. J. Rob. Res.*, 2(3), 3-21. doi: 10.1177/027836498300200301.

Karim, A., Hitzer, J., Lechler, A. & Verl, A. (2017). Analysis of the dynamic behavior of a six-axis industrial robot within the entire workspace in respect of machining tasks. *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 670-675. doi: 10.1109/aim.2017.8014094.

Khalil, W. & Creusot, D. (1997). SYMORO : A system for the symbolic modelling of robots. *Robotica*, 15(2), 153-161. doi: 10.1017/S0263574797000180.

Khalil, W. & Dombre, E. (2004). *Modeling, Identification and Control of Robots*. Butterworth-Heinemann, Oxford.

Khalil, W., Boyer, F. & Morsli, F. (2017). General Dynamic Algorithm for Floating Base Tree Structure Robots With Flexible Joints and Links. *Journal of Mechanisms and Robotics*, 9(3). doi: 10.1115/1.4035798.

Kim, J. & Croft, E. (2018). Preshaping input trajectories of industrial robots for vibration suppression. *Rob. Comput. Integr. Manuf.*, 54, 35-44. doi: 10.1016/j.rcim.2018.05.009.

Kumar, S., Szadkowski, K., Mueller, A. & Kirchner, F. (2020). An Analytical and Modular Software Workbench for Solving Kinematics and Dynamics of Series-Parallel Hybrid Robots. *Journal of Mechanisms and Robotics*, 12(2). doi: 10.1115/1.4045941.

Kurz, T., Eberhard, P., Henninger, C. & Schiehlen, W. (2010). From Neweul to Neweul-M2: symbolical equations of motion for multibody system analysis and synthesis. *Multibody Sys.Dyn.*, 24(1), 25-41. doi: 10.1007/s11044-010-9187-x.

Lessard, J., Swiatek, G., Liu, Z. & Hazel, B. (2011). Flexible Multibody Dynamics and Effects of Nonlinear Joint stiffness for Vibration Analysis in a Robotic Grinding Process. *ECCOMAS-Thematic Conference on Multibody Dynamics*.

Li, Y., Liu, X., Peng, Z. & Liu, Y. (2002). The identification of joint parameters for modular robots using fuzzy theory and a genetic algorithm. *Robotica*, 20(5), 509–517. doi: 10.1017/S0263574702004277.

Lightcap, C. & Banks, S. (2007). Dynamic identification of a mitsubishi pa10-6ce robot using motion capture. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3860-3865. doi: 10.1109/IROS.2007.4399425.

Luh, J., Walker, M. & Paul, R. (1980). On-line computational scheme for mechanical manipulators. *J. Dyn. Syst. Meas. Contr.*, 102, 69-76. doi: 10.1115/1.3149599.

Lynch, A. G. & Vanderploeg, M. J. (1995). A Symbolic Formulation for Linearization of Multibody Equations of Motion. *J. Mech. Des.*, 117(3), 441-445.

Maamar, A., Gagnol, V., Le, T.-P. & Sabourin, L. (2020). Pose-dependent modal behavior of a milling robot in service. *The International Journal of Advanced Manufacturing Technology*, 107(1), 527-533. doi: 10.1007/s00170-020-04974-y.

Maia, N. M. M. & Silva, J. M. M. (Eds.). (1997). *Theoretical and Experimental Modal Analysis*. Research Studies Press.

MathWorks. (2020b). MATLAB Documentation, Global Optimization Toolbox. Natick, Massachusetts, USA: The Mathworks, Inc.

Mejri, S., Gagnol, V., Le, T.-P., Sabourin, L., Ray, P. & Paultre, P. (2013). Experimental protocol for the dynamic modeling of machining robots. *21 Congres Français De Mecanique, CFM*, pp. 1-6.

Mejri, S., Gagnol, V., Le, T.-P., Sabourin, L., Ray, P. & Paultre, P. (2016). Dynamic characterization of machining robot and stability analysis. *The International Journal of Advanced Manufacturing Technology*, 82(1), 351-359. doi: 10.1007/s00170-015-7336-3.

Miranda-Colorado, R. & Moreno-Valenzuela, J. (2018). Experimental parameter identification of flexible joint robot manipulators. *Robotica*, 36(3), 313–332. doi: 10.1017/S0263574717000224.

Moberg, S. (2010). *Modeling and control of flexible manipulators*. (Ph.D. thesis, Linköping University).

Moberg, S. & Hanssen, S. (2008). On Feedback Linearization for Robust Tracking Control of Flexible Joint Robots. *IFAC Proceedings Volumes*, 41(2), 12218-12223. doi: 10.3182/20080706-5-KR-1001.02069. 17th IFAC World Congress.

Mottershead, J. E. & Ram, Y. M. (2006). Inverse eigenvalue problems in vibration absorption: Passive modification and active control. *Mechanical Systems and Signal Processing*, 20(1), 5-44. doi: 10.1016/j.ymssp.2005.05.006.

Mousavi, S., Gagnol, V., Bouzgarrou, B. C. & Ray, P. (2017). Dynamic modeling and stability prediction in robotic machining. *The International Journal of Advanced Manufacturing Technology*, 88(9), 3053 - 3065.

Müller, A. (2017, May). Recursive second-order inverse dynamics for serial manipulators. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2483-2489. doi: 10.1109/ICRA.2017.7989289.

Neubauer, M., Gattringer, H., Müller, A., Steinhauser, A. & Höbarth, W. (2015). A two-stage calibration method for industrial robots with joint and drive flexibilities. *Mechanical Sciences*, 6(2), 191-201. doi: 10.5194/ms-6-191-2015.

Ohr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S. & Sander, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. *Proceedings of ISMA2006: International Conference on Noise and Vibration Engineering*, pp. 3305-3313.

Ott, C. (2008). *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Springer-Verlag.

Palomba, I. & Vidoni, R. (2019). Flexible-Link Multibody System Eigenvalue Analysis Parameterized with Respect to Rigid-Body Motion. *Applied Sciences*, 9(23). doi: 10.3390/app9235156.

Park, F., Kim, B., Jang, C. & Hong, J. (2018). Geometric algorithms for robot dynamics: a tutorial review. *Appl. Mech. Rev.*, 70(1), 1-18. doi: 10.1115/1.4039078.

Pham, M.-N., Hamelin, P., Hazel, B. & Liu, Z. (2020). A Two-Stage State Feedback Controller Supported by Disturbance-State Observer for Vibration Control of a Flexible-Joint Robot. *Robotica*, 38(6), 1082-1104. doi: 10.1017/S0263574719001267.

Pratt, G. & Williamson, M. (1995). Series elastic actuators. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 399-406.

Quintana, G. & Ciurana, J. (2011). Chatter in machining processes: A review. *Int. J. Mach. Tools Manuf*, 51(5), 363-376. doi: 10.1016/j.ijmachtools.2011.01.001.

Reynders, E., Houbrechts, J. & De Roeck, G. (2012). Fully automated (operational) modal analysis. *Mech. Syst. Sig. Process.*, 29, 228-250. doi: 10.1016/j.ymssp.2012.01.007.

Samin, J. & Fisette, P. (2003). *Symbolic Modeling of Multibody Systems*. Kluwer Academic Publishers.

Schiehlen, W. (Ed.). (1990). *Multibody System Handbook*. Springer-Verlag.

Schiehlen, W. (Ed.). (1993). *Advanced Multibody System Dynamics Simulation and Software Tools*. Springer Netherlands.

Schiehlen, W. (1997). Multibody System Dynamics: Roots and Perspectives. *Multibody Sys.Dyn.*, 1, 149-188. doi: 10.1023/A:1009745432698.

Schiehlen, W. & Erberhard, P. (2014). *Applied Dynamics*. Springer.

Schmitke, C. & Goossens, P. (2011). Symbolic computation techniques for multibody model development and code generation. *ECCOMAS Thematic Conference on Multibody Dynamics*, pp. 1-7.

Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. (2009). *Robotics Modelling, Planning and Control*. Springer.

Spong, M. W. (1987). Modeling and control of elastic joint robots. *J. Dyn. Syst. Meas. Contr.*, 109(4), 310-318. doi: 10.1115/1.3143860.

Sun, J., Zhang, W. & Dong, X. (2020). Natural Frequency Prediction Method for 6R Machining Industrial Robot. *Applied Sciences*, 10(22). doi: 10.3390/app10228138.

Swevers, J., Ganseman, C., De Schutter, J. & Van Brussel, H. (1996). Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, 10(5), 561-577. doi: 10.1006/mssp.1996.0039.

Swevers, J., Verdonck, W. & De Schutter, J. (2007). Dynamic Model Identification for Industrial Robots. *IEEE Control Systems Magazine*, 27(5), 58-71. doi: 10.1109/MCS.2007.904659.

Tomei, P. (1990). An observer for flexible joint robots. *IEEE Transactions on Automatic Control*, 35(6), 739-743.

Tomei, P. (1991). A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control*, 36(10), 1208-1213. doi: 10.1109/9.90238.

Troger, H. & Steindl, A. (1991). *Nonlinear Stability and Bifurcation Theory An Introduction for Engineers and Applied Scientists*. Springer-Verlag Wien.

Tsai, L.-W. (2003). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley-Interscience.

Uchida, T. (2011). *Real-time Dynamic Simulation of Constrained Multibody Systems using Symbolic Computation*. (Ph.D. thesis, University of Waterloo).

Vu, V.-H., Thomas, M., Lakis, A. & Marcouiller, L. (2011). Operational modal analysis by updating autoregressive model. *Mech. Syst. Sig. Process.*, 25(3), 1028-1044. doi: 10.1016/j.ymssp.2010.08.014.

Vu, V.-H., Thomas, M., Lafleur, F. & Marcouiller, L. (2013). Towards an automatic spectral and modal identification from operational modal analysis. *J. Sound Vib.*, 332(1), 213-227. doi: 10.1016/j.jsv.2012.08.019.

Vu, V.-H., Liu, Z., Thomas, M., Li, W. & Hazel, B. (2016). Output-only identification of modal shape coupling in a flexible robot by vector autoregressive modeling. *Mech. Mach. Theory*, 97, 141-154. doi: 10.1016/j.mechmachtheory.2015.11.005.

Vu, V.-H., Liu, Z., Thomas, M. & Hazel, B. (2017). Modal analysis of a light-weight robot with a rotating tool installed at the end effector. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(9), 1664-1676. doi: 10.1177/0954406215619451.

Wang, Y., Belzile, B., Angeles, J. & Li, Q. (2019). The Modeling of Redundantly Actuated Mechanical Systems. *Journal of Mechanisms and Robotics*, 11(6). doi: 10.1115/1.4044540.

Wernholt, E. & Moberg, S. (2011). Nonlinear gray-box identification using local models applied to industrial robots. *Automatica*, 47(4), 650-660. doi: https://doi.org/10.1016/j.automatica.2011.01.021.

Wittkopf, A. (2008). Automatic code generation and optimization in Maple. *Journal of Numerical Analysis, Industrial and Applied Mathematics*, 3(1-2), 167-180.

Wu, J., Wang, J. & You, J. (2010). An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5), 414-419. doi: doi.org/10.1016/j.rcim.2010.03.013.

Yang, K., Yang, W. & Wang, C. (2018). Inverse dynamic analysis and position error evaluation of the heavy-duty industrial robot with elastic joints: an efficient approach based on Lie group. *Nonlinear Dyn.*, 24, 487-504. doi: 10.1007/s11071-018-4205-2.

Zaghbani, I., Songmene, V. & Bonev, I. (2013). An experimental study on the vibration response of a robotic machining system. *Journal of Engineering Manufacture*, 227(6), 866-880. doi: 10.1177/0954405413477067.

Zollo, L., Lopez, E., Spedaliere, L., Aracil, N. & Guglielmelli, E. (2015). Identification of Dynamic Parameters for Robots with Elastic Joints. *Advances in Mechanical Engineering*, 7(2), 1-15. doi: 10.1155/2014/843186.