POLITECNICO DI TORINO

Master's Degree in Mechanical Engineering

Flexible Multibody Simulation Framework for a Single 6-RUS Leg for Space-Docking Applications

THIS THESIS WAS CONDUCTED IN COLLABORATION WITH CHALMERS UNIVERSITY OF TECHNOLOGY

Supervisors:
Enrico Galvagno
Shivesh Kumar
Petri Piiroinen

Candidate:
Piero Tufilli

Academic Year 2024/2025

Abstract

Space robotics increasingly relies on compliant manipulation for safe, low-impact docking under pose uncertainty. PACOMA is a 6-RUS (Revolute–Universal–Spherical) parallel continuum manipulator designed for low-impact docking. This thesis develops a control-oriented flexible multibody framework at the leg level, modeling a planar surrogate as a four-bar mechanism with two rigid links and one flexible Timoshenko beam. The beam is discretized via the finite element method and coupled to the rigid links through holonomic constraints, yielding differential-algebraic equations implemented in MATLAB and integrated with a generalized-α scheme. Two self-contained verification blocks support the framework: a compact rigid four-bar baseline and an isolated beam module. These are used for symbolic EOM checks and static/modal validations, as well as preliminary time-domain comparisons within a commercial multibody dynamics environment (MSC Adams). A leg-level rigid–flexible–rigid prototype is also assembled in MSC Adams to mirror the MATLAB formulation; quantitative cross-validation is deferred until the coupled model attains stable convergence. The resulting leg-level simulation pipeline exposes clear interfaces for state-space extraction and is structured for model-based control, with particular attention to LQR-type design in subsequent development. While comprehensive end-effector load cases and closed-loop tests on the coupled flexible model are reserved for future work, the thesis consolidates the modeling stack needed to scale from leg-level analysis to the full 6-RUS architecture. The framework aligns with PACOMA's goals—safe interaction through compliant limbs and robustness to docking misalignment—and outlines a practical roadmap to complete flexible coupling and transition from simulation to control.

Table of Contents

List of Tables V

List of Figures VI

1 Introduction 1

1.1 Motivation and Background 1

1.2 PACOMA at a Glance 2

1.3 From PACOMA to a Control-Oriented Leg-Level Planar Simplified Model 3

1.4 Modeling Assumptions 4

1.5 Thesis Roadmap 5

1.6 Software used: MATLAB and ADAMS View 5

2 Rigid four–bar model 8

2.1 Geometry and generalized coordinates 11

2.2 Kinematics, energies, and open–chain dynamics 13

2.3 Loop closure and constraint equations 14

2.4 Implicit constrained dynamics (KKT, index–3 DAE) 14

2.5 Explicit geometric reduction (single ODE) 16

2.6 Initial conditions and consistency 17

2.7 Numerical behavior and comparison 17

3 Timoshenko Beam 21

3.1 Kinematics and Strains 21

3.2 Strong Form and Boundary Conditions 25

3.3 Weak Form and Two-Node Timoshenko Element 26

3.4 Loads, Damping, and the Semi-Discrete Equations 30

3.5 Time Integration: Newmark and Generalized-α 31

3.5.1 Newmark's family. 31

3.5.2 Generalized-α scheme. 31

3.6 Modal Analysis and Worked Examples 31

3.6.1 The generalized eigenvalue problem. 31
3.6.2 Orthogonality and normalization. 32
3.7 Rayleigh Damping Identification 32
3.8 Validation with a Cantilever Beam Test Case 32
3.9 Numerical results 34
3.10 Natural frequencies 39

4 Simplified Flexible Multi Body Simulation Model of PACOMA Leg Mechanism 41
4.1 Geometry, coordinates and finite element mesh 44
4.2 Energies in the first rigid–flexible formulation 47
4.2.1 Rigid links 47
4.2.2 Flexible link: kinetic energy 47
4.2.3 Flexible link: gravitational and elastic energy 48
4.2.4 Lagrangian and equations of motion 48
4.3 Rigid loop–closure constraint and its consequences 49
4.4 Time integration and damping in the first DAE solver 50
4.4.1 Generalized-α scheme with fixed tangent 50
4.4.2 Rayleigh damping and its limitations 50
4.5 Observed behavior and mismatch with the target mechanism 51
4.6 Key differences with the final rigid–flexible formulation 52

5 Full Flexible Multi Body Simulation Model of PACOMA Leg Mechanism 56
5.1 Aim and modeling scope 56
5.2 Geometry, coordinates, and structure of the unknowns 64
5.3 Modeling assumptions and rationale 66
5.4 Mesh, active degrees of freedom, and numbering 71
5.5 Co-rotational kinematics of the beam 71
5.6 Strain measures and constitutive density 73
5.6.1 Modeling remarks. 74
5.7 Energies and their physical content 74
5.8 Euler–Lagrange operator, mass matrix, and non-inertial term 76
5.9 Element matrices and global assembly 78
5.10 Loop-closure and constrained equations 80
5.10.1 Acceleration-level stabilization (Baumgarte). 81
5.10.2 Practical checks. 83
5.11 Tendon model, endpoint Jacobians, and static pretension 85
5.11.1 Spring–damper along the cable line. 86
5.11.2 Static pretension problem. 86
5.12 Time integration: generalized-α scheme with constraints 93

5.13 Structural damping and selection of Rayleigh coefficients 96
5.14 Numerical conditioning and practical safeguards 97
    5.14.1 Scaling 97
    5.14.2 Constraint Jacobian 98
    5.14.3 Consistent linearizations 98
    5.14.4 Trust region on u, v 98
    5.14.5 Mild regularization 98
    5.14.6 Stabilization coefficients 98
5.15 Verification strategy 98
5.16 Simulations 99

6 Conclusions and Future Work 103
6.1 Summary of contributions 103
6.2 Limitations and possible extensions 103
6.3 LQR control in a nutshell 105
    6.3.1 1) Pick an operating point. 105
    6.3.2 2) Obtain a small linear model. 105
    6.3.3 3) Choose the LQR weights. 105
    6.3.4 4) Solve the Riccati equation and form the gain. 106
    6.3.5 5) Keep it gentle with flexible modes. 106
    6.3.6 6) Test and iterate. 106
6.4 From one leg to a six–leg parallel arrangement 106
6.5 Concluding remarks 108

Bibliography 110

List of Tables

1.1 Legend for Fig. 1.2. 3

3.1 Symbols, units, and conventions used in the Timoshenko beam chapter. 22
3.2 Comparison of the first three natural frequencies for the test-case cantilever beam 34
3.3 Fixed geometric, material, and numerical parameters used in the Timoshenko beam study. 34
3.4 Comparison of final static displacements with and without gravity . 36
3.5 Modal frequencies of the beam 36

4.1 Symbols used in the simplified flexible multi body simulation model chapter. 41
4.2 Fixed geometric, material, and numerical parameters used in the simplified flexible multi body simulation model. 46

5.1 Symbols used in the full flexible multi body simulation model chapter. 57
5.2 Fixed geometric, material, and numerical parameters used in the full flexible multi body simulation model. 65
5.3 Element lengths and axial strains in the static pretension solution of a four-element test mesh. The first three elements are in mild tension, while almost the entire compression is localized in the last element, which exhibits an unrealistically large negative strain. . . . 92

List of Figures

1.1 Docking scenario with PACOMA placed between two spacecraft [1, 2]. The compliant legs mitigate contact forces and accommodate misalignment during capture. 1
1.2 PACOMA overview: (a) CAD design; (b–d) prototype sub-assemblies [1]. 2
1.3 Numbered sketch of one PACOMA leg. 1: fixed revolute at the base. 2: fixed revolute at the end effector (EE). 3: lower rigid link (base link). 4: flexible beam. 5: upper rigid link (EE link). 6: revolute joint between lower rigid link and flexible beam. 7: revolute joint between flexible beam and upper rigid link. 4

2.1 Geometry of the rigid four–bar. 12
2.2 Joint angles: explicit ODE (solid blue) vs. commercial reference (dashed red). 18
2.3 Angular velocities: explicit ODE (solid blue) vs. commercial reference (dashed red). 18
2.4 Joint angles: implicit DAE (blue) vs. explicit ODE (red). The implicit solution shows the expected amplitude reduction due to high–frequency damping. 19
2.5 Angular velocities: implicit DAE (blue) vs. explicit ODE (red). The damping trend is consistent across all joints. 19
2.6 Animation of MATLAB simulation with explicit solver (only gravity force applied) 20
2.7 Animation of ADAMS simulation (only gravity force applied) 20

3.1 Two-node Timoshenko element and its nodal degrees of freedom $[u, v, \omega_z]$. 24
3.2 Local coordinate system (left) – global Cartesian coordinate system (right) 27
3.3 Linear two-node shape functions on $\xi \in [-1,1]$. 28
3.4 Timoshenko beam geometry and boundary conditions used throughout. 33

3.5 Displacement-Velocity-Acceleration of the last node (MATLAB and Adams results) . . . 37
3.6 Comparison of damping results between MATLAB and MSC Adams 37
3.7 Animation of MATLAB simulation (only gravity force and vertical end-node force applied). . . . 38
3.8 Animation of ADAMS simulation (only gravity force and vertical end-node force applied). . . . 38
3.9 Representation of the first frequency: 186.5 Hz . . . 39
3.10 Representation of the second frequency: 1177.6 Hz . . . 39
3.11 Representation of the third frequency: 3337.8 Hz . . . 40
3.12 Representation of the fourth frequency: 6659.7 Hz . . . 40
3.13 Representation of the fifth frequency: 8656.8 Hz (the beam lengthens and shortens linearly) . . . 40

4.1 Rigid-flexible model's initial geometry. . . . 45
4.2 Trend of the angles of the two rigid links and the flexible beam (only gravity force applied) . . . 53
4.3 Longitudinal and transverse deformations of the middle node (only gravity force applied) . . . 54
4.4 Animation of MATLAB simulation (only gravity force applied) . . . 54
4.5 Animation of ADAMS simulation (only gravity force applied) . . . 55

5.1 Sketch of the desired PACOMA leg after pretension. . . . 57
5.2 Rigid-flexible model's initial geometry. . . . 65
5.3 Rigid-flexible model's active degrees of freedom. Flexible beam discretized with 2 elements just as example. . . . 67
5.4 Qualitative rendering of the pre-tensioned leg (MSC Adams). The cable runs from a point near the tip of link 1 to the flexible–beam tip, generating an outward bend under pretension. Colors indicate bending intensity; red bars are the rigid links. . . . 87
5.5 Angles q_1, q_2 and q_3 during the pretensioning transient. . . . 87
5.6 The displacement of the three internal nodes of the flexible beam (when it is divided into 4 elements) during the pretensioning transient. 88
5.7 Spring deformation during the pretensioning transient. . . . 88
5.8 Spring force during the pretensioning transient. . . . 89
5.9 Static pretension configuration predicted by the current co-rotational Timoshenko model. . . . 91
5.10 Animation of MATLAB simulation (free-fall after pretension). . . . 100
5.11 Animation of ADAMS simulation (free-fall after pretension). . . . 100
5.12 Angles q_1, q_2 and q_3 during the 3 seconds free-fall simulation after the pretensioning transient. . . . 101

5.13 The displacement of the three internal nodes of the flexible beam (when it is divided into 4 element) during the 3 seconds free-fall simulation after the pretensioning transient. 101
5.14 Spring deformation during the 3 seconds free-fall simulation after the pretensioning transient. 102
5.15 Spring force during the 3 seconds free-fall simulation after the pretensioning transient. 102

Listings

2.1 Implicit generalized-α step (KKT DAE). 15
2.2 Explicit reduction: right-hand side for the scalar ODE. 16
5.1 Constraint projection at position and velocity levels (`project_loop`). 68
5.2 Co-rotational reconstruction of current beam-node positions (`current_geom_from_q`). 70
5.3 Loop-closure maps: position constraint `phi(q)` and Jacobian `Jc(q)`. 83
5.4 Tendon generalized force `Qten` and elastic tangent `Kten`. 90
5.5 One generalized-α Newton corrector with constraints (direct KKT solve). 95

Chapter 1

Introduction

1.1 Motivation and Background

Robots for space applications need to be safe during contact, robust to pose errors, and light on the base during docking and servicing. Serial arms are flexible to use, but add mass and inertia along the chain. This can increase impact forces and makes thermal and dust protection harder. Parallel robots are stiffer and more accurate thanks to the closed kinematic loops, but if they are fully rigid they can still create large forces when the pose is wrong.

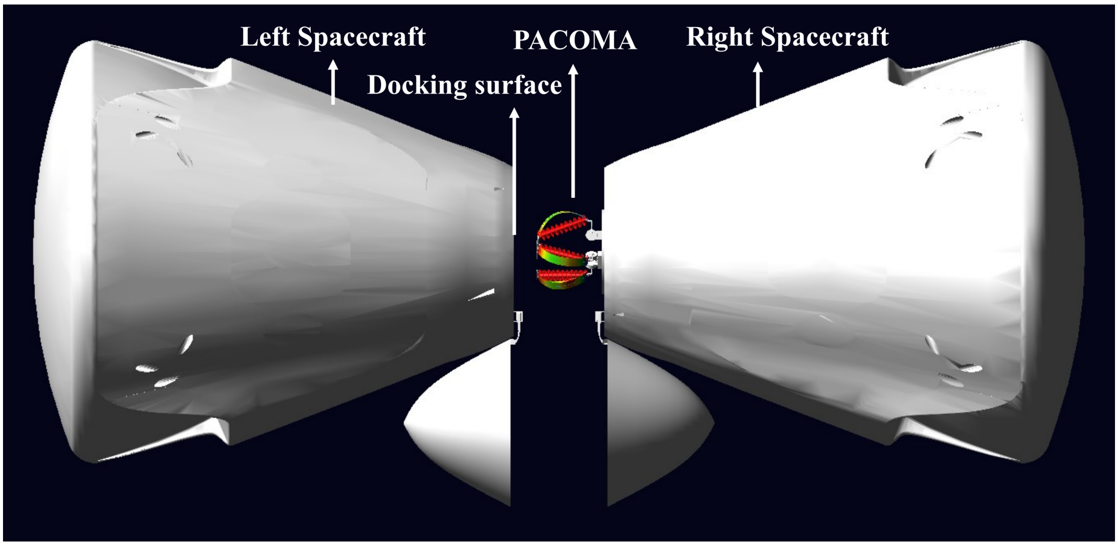


Figure 1.1: Docking scenario with PACOMA placed between two spacecraft [1, 2]. The compliant legs mitigate contact forces and accommodate misalignment during capture.

Parallel continuum robots replace some rigid parts with flexible elements. This adds passive compliance, so the robot can tolerate misalignment and keep contact forces low. PACOMA is a 6-RUS parallel continuum robot for docking [1, 2] and similar tasks in space. RUS means that each leg has a Revolute, a Universal, and a Spherical joint. The end effector (EE) connects to the base through six compliant legs. The motors and most sensors sit on the lower base to simplify thermal control and protect the hardware.

1.2 PACOMA at a Glance

Each of the six legs has four main parts: (i) a proximal rigid link driven by a base actuator, (ii) a universal joint, (iii) a leaf-spring unit with a pre-tensioned tendon and a damper, (iv) a spherical joint at the EE [1]. Most of the compliance comes from the leaf-spring–tendon unit. The tendon adds pretension so the spring keeps a stable curved shape; the damper reduces rebound.

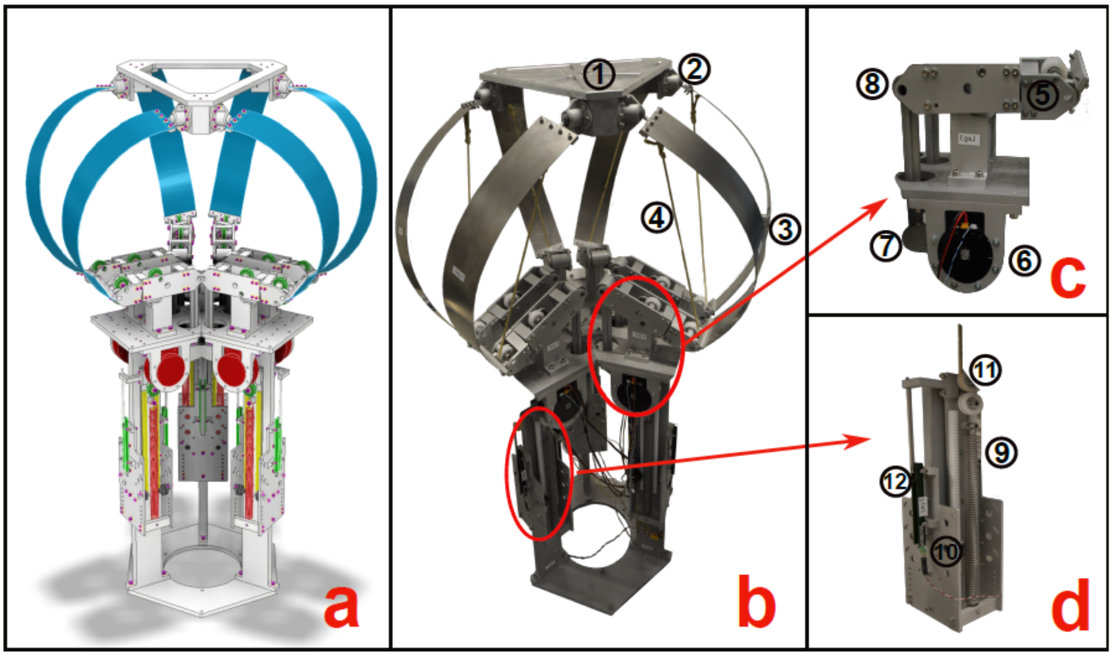


Figure 1.2: PACOMA overview: (a) CAD design; (b–d) prototype sub-assemblies [1].

A typical prototype uses titanium leaf springs (rectangular section, about 1.2 mm thick, 77 mm wide, and about 500 mm long, as in the PACOMA prototype [1]), a pre-load of about 100 mm, and quasi-direct-drive actuators with a working range near ±20° through a small parallelogram linkage. The pretension–damper unit provides damping around $15\,\mathrm{N\,s\,cm^{-1}}$ and includes a linear sensor on the tendon path. In tests and simulations, PACOMA shows smooth motion and low-impact contacts during docking. For example, with a misalignment of about 7.5° the peak contact force is around 45.5 N [1].

No.	Component	Role / Notes
1	End-effector	Interface to docking surface / payload.
2	Spherical joint	EE-side passive orientation.
3	Titanium leaf spring	Main compliant element (continuum leg).
4	Pretension tendon	Applies pre-load to set curvature.
5	Universal joint	Intermediate passive joint (U).
6	Actuator	Base-mounted drive (QDD).
7	Revolute joint	Motor output joint (R).
8	Parallelogram linkage	Motion transmission / range shaping.
9	Coil spring (pretension)	Provides pretension on the tendon path.
10	Damper	Dissipates energy, smooths contact.
11	Pulley set	Guides tendon path/mechanics.
12	Linear potentiometer	Measures tendon travel distance.

Table 1.1: Legend for Fig. 1.2.

1.3 From PACOMA to a Control-Oriented Leg-Level Planar Simplified Model

Modeling all six compliant legs in full 3D is the end goal, but for control design a simpler step is adopted at the initial stage. This thesis therefore develops and evaluates a 2D leg-level model, referred to as the planar simplified model. It keeps the key rigid–flexible effects that matter for docking, but is lighter to analyze and control.

Concretely, a single compliant leg is represented within a minimal four-bar configuration consisting of two rigid links and one flexible link:

- the flexible link is a Timoshenko beam,
- the beam is discretized with finite elements,
- holonomic constraints enforce the joints between rigid and flexible parts and the loop closure,
- the equations of motion are differential–algebraic and we integrate them with a generalized-α scheme.

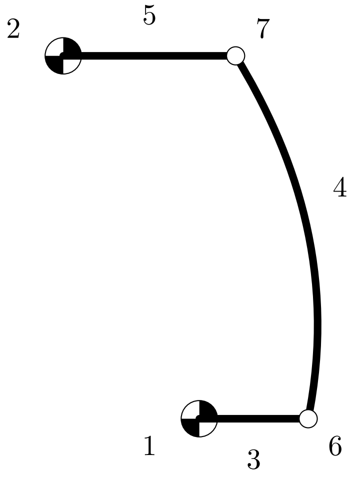


Figure 1.3: Numbered sketch of one PACOMA leg. 1: fixed revolute at the base. 2: fixed revolute at the end effector (EE). 3: lower rigid link (base link). 4: flexible beam. 5: upper rigid link (EE link). 6: revolute joint between lower rigid link and flexible beam. 7: revolute joint between flexible beam and upper rigid link.

This model is enough to study compliance, how energy moves between rigid and flexible parts, and the interface forces at the joints. There have been used two small, stand-alone blocks to check and validate the work: (i) a compact rigid four-bar baseline, and (ii) an isolated flexible beam module. These help with symbolic checks, static and modal tests, and first comparisons with a commercial multibody tool (MSC Adams).

1.4 Modeling Assumptions

We keep the model simple with a few assumptions:

1. Planar reduction: the leg is studied in 2D. The motion is projected onto a plane that captures the main deformation relevant to docking.
2. Flexible beam: the flexible link follows Timoshenko beam theory (it includes shear deformation and rotary inertia). Linear finite elements are used; each node has three degrees of freedom: axial displacement u, transverse displacement v, and rotation θ.
3. Rigid links: the proximal and distal links are rigid bodies with lumped mass and inertia. Clearances and friction in the joints are neglected.
4. Coupling: kinematic compatibility is applied between rigid and flexible parts with holonomic constraints. External loads can act on the end effector (EE).

5. Time integration and damping: the equations are integrated with the generalized-α scheme. Material and numerical damping are set with explicit parameters and reported in the simulations.

Choice of Timoshenko beam model. In previous PACOMA-related work, the compliant legs were modeled as Cosserat rods, providing a geometrically exact description that can capture all deformation modes of a three-dimensional slender body [2]. In that context, however, the analysis was restricted to a kinetostatic setting and focused on equilibrium configurations of the continuum structure rather than on its full dynamic response. In contrast, the goal of this thesis is to derive and simulate a dynamic rigid–flexible leg model, with explicit inertial coupling between the rigid links and the flexible beam, and to integrate it in a time-stepping pipeline suitable for control-oriented studies. For this reason a planar Timoshenko beam model is adopted as a compromise between fidelity and algorithmic complexity: it retains shear deformation and rotary inertia, matches well-established finite-element formulations in structural dynamics, and leads to sparse mass and stiffness matrices that are easier to handle within the generalized-α time integrator used throughout the work. The Cosserat formulation remains more general, but the Timoshenko model is sufficiently rich for the planar leg-level dynamics considered here, while being simpler to implement and to interface with standard rigid multibody solvers.

1.5 Thesis Roadmap

This introduction motivates the leg-level surrogate of PACOMA as a stepping stone toward the full 6-RUS architecture [1, 2]. The remainder of the thesis follows the plan in the chapter outline: a brief baseline on the rigid four-bar, the flexible beam module and its modal/static validation, the coupled rigid–flexible four-bar with DAE and generalized-α integration, and concluding perspectives toward leg-to-platform scaling and model-based control (LQR).

1.6 Software used: MATLAB and ADAMS View

This thesis uses two tools. MATLAB is the main environment [3] for modeling, equation assembly, and simulation. ADAMS View serves as an external reference [4] for cross-checks.

MATLAB (modeling and simulation)

The planar simplified model of one leg is implemented in MATLAB. The flexible link is built with the finite element method (Timoshenko beam [5]) and coupled to the two rigid links through holonomic constraints. The differential–algebraic equations are integrated with a generalized-α scheme [6]. Plots, mode shapes, and static checks are produced in MATLAB.

ADAMS View (external reference)

Three reference models are built in ADAMS View to mirror the MATLAB setup:

- a rigid four-bar baseline,
- an isolated flexible beam with FE Part (same length, section, and material data),
- a leg-level rigid–flexible–rigid assembly matching the planar simplified model.

ADAMS is used only to check shapes, static deflection, and basic time responses under the same geometry and loads. Later chapters refer back to this section when comparisons are shown.

Consistency rules for comparisons

To keep MATLAB and ADAMS aligned:

- the same units and gravity setting are used,
- rigid-link geometry and mass properties are matched,
- beam data (length, cross section, E, G, density) are matched and the mesh is chosen to obtain comparable mode shapes,
- the joint layout and constraints mirror the MATLAB model,
- initial conditions and load cases are replicated,
- outputs are taken at the same points (beam ends and joint angles).

Scope and limits

ADAMS is used only for cross-checks. Tool-specific features that are not part of the thesis model are avoided. Any necessary differences in numerical settings (integrator or tolerances) are noted alongside each comparison.

Code availability

All MATLAB scripts, symbolic generation files, and ADAMS input decks developed for this thesis are available in a GitLab repository hosted at Chalmers:

`https://git.chalmers.se/m2/dynamics/rail/constrained_rigid_flexible_modeling/master-thesis-piero`

The repository contains the rigid four–bar baseline, the isolated flexible beam models, and the full rigid–flexible leg model used in the simulations presented in the following chapters.

Chapter 2

Rigid four–bar model

This chapter builds a rigid four–bar mechanism and uses it as a baseline before adding a flexible beam. The baseline has three goals: (i) check that the symbolic equations of motion are correct; (ii) compare two ways to enforce the closed kinematic loop—an *implicit* constrained formulation written as a Karush–Kuhn–Tucker (KKT) saddle–point system that yields a differential–algebraic equation (DAE), and an *explicit* geometric reduction that produces a single ordinary differential equation (ODE); (iii) verify that the simulated motion matches a commercial multibody reference when the physical data and tolerances are aligned. Such a comparison between implicit DAE and explicit reduced ODE formulations is common in constrained multibody dynamics; see, for instance, [7] for a related discussion in a different robotic setting. Parts of the MATLAB structure follow standard practice; many steps were inspired by an internship report and then rewritten to fit the geometry and solvers used here [8].

Table 2.1: Symbols used in the rigid four–bar model chapter.

Symbol	**Description**	**Units (SI)**
O, C	Ground revolute joints (fixed pivots of bars 1 and 3).	m
A, B	Moving revolute joints connecting the three bars.	m
$\mathbf{r}_{OC}$	Ground offset vector from O to C in the global frame.	m
L_1, L_2, L_3	Lengths of bars 1, 2, and 3, respectively.	m
i	Bar index, with $i \in \{1,2,3\}$.	–
m_i	Mass of bar i.	kg

Continued on next page

Table 2.1: Symbols used in the rigid four–bar model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
I_i	Planar mass moment of inertia of bar i about its centre of mass.	$\mathrm{kg\,m^2}$
$\mathbf{p}_{G1}, \mathbf{p}_{G2}, \mathbf{p}_{G3}$	Position vectors of the centres of mass of bars 1, 2, and 3 in the global frame.	m
$\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$	Generalized coordinate vector (joint angles of the three bars).	rad
q_1	Absolute angle of bar 1 about the ground joint O.	rad
q_2	Relative angle of bar 2 with respect to bar 1 at joint A.	rad
q_3	Absolute angle of bar 3 about the ground joint C.	rad
$\dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Generalized velocity and acceleration vectors.	rad/s, rad/$\mathrm{s^2}$
T	Total kinetic energy of the rigid four–bar mechanism.	J
U	Gravitational potential energy.	J
$L = T - U$	Lagrangian of the rigid four–bar system.	J
g	Gravitational acceleration, acting along the negative y–direction.	$\mathrm{m/s^2}$
$\mathbf{M}(\mathbf{q})$	3×3 mass matrix of the open–chain system, expressed in generalized coordinates.	$\mathrm{kg\,m^2}$
$\mathbf{c}(\dot{\mathbf{q}}, \mathbf{q})$	Vector of gravity and velocity–dependent terms in the open–chain equations of motion.	N m
$\boldsymbol{\tau}$	Vector of applied generalized torques acting on the three joints.	N m
$\boldsymbol{\phi}(\mathbf{q})$	Vector of position–level loop–closure constraints enforcing that joints A and B meet the tip of bar 3.	m
$\mathbf{K}(\mathbf{q})$	Constraint Jacobian matrix, i.e., the Jacobian of $\boldsymbol{\phi}(\mathbf{q})$ with respect to $\mathbf{q}$ (sometimes denoted $\mathbf{J}_c$ in the literature).	m
$\mathbf{k}(\dot{\mathbf{q}}, \mathbf{q})$	Right–hand side of the acceleration–level constraint equation, defined from $-\dot{\mathbf{K}}(\dot{\mathbf{q}}, \mathbf{q})\,\dot{\mathbf{q}}$.	$\mathrm{m/s^2}$

Continued on next page

Table 2.1: Symbols used in the rigid four–bar model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers associated with the loop–closure constraints (constraint reactions).	N
$\mathbf{H}$	Vector of generalized forces entering the residual of the equations of motion.	N, N m
$\mathbf{K}_t$	Tangent stiffness matrix, defined as $\partial\mathbf{H}/\partial\mathbf{q}$.	N m
$\mathbf{C}_t$	Tangent damping matrix, defined as $\partial\mathbf{H}/\partial\dot{\mathbf{q}}$.	N m s
$\mathbf{r}_q$	Residual of the equations of motion in the KKT system.	N, N m
$\mathbf{r}_\lambda$	Residual of the position constraints, i.e., $\boldsymbol{\phi}(\mathbf{q})$.	m
$\mathbf{S}$	Effective tangent (iteration) matrix used in the Newton step of the generalized–α scheme.	–
α_m, α_f	Generalized–α integration parameters controlling numerical dissipation.	–
β, γ	Newmark–type integration parameters derived from α_m and α_f.	–
β_p, γ_p	Effective parameters multiplying $\mathbf{M}$ and $\mathbf{C}_t$ in the generalized–α iteration matrix.	–
ρ_∞	Target spectral radius at infinity used to select α_m and α_f.	–
Δt	Time step size for time integration.	s
y	Independent generalized coordinate used in the explicit geometric reduction (chosen as $y = q_1$).	rad
$\boldsymbol{\gamma}_j(y)$	Geometric mapping from the scalar coordinate y to the full configuration $\mathbf{q}$ on branch $j \in \{1,2\}$.	rad
$\mathbf{G}(y)$	Reduction Jacobian matrix $\partial\boldsymbol{\gamma}_j/\partial y$.	–
$\mathbf{g}(y, \dot{y})$	Kinematic bias term defined as $\dot{\mathbf{G}}(y)\,\dot{y}$.	rad/s
$M_y(y)$	Scalar reduced mass term in the explicit single–DOF ODE.	kg m^2
$c_y(\dot{y}, y)$	Reduced generalized force term (gravity and inertial contributions) in the explicit ODE.	N m

Continued on next page

Table 2.1: Symbols used in the rigid four–bar model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\tau_y(y)$	Reduced generalized torque appearing in the explicit ODE.	N m
$\mathbf{q}_0, \dot{\mathbf{q}}_0$	Initial generalized position and velocity, chosen to satisfy the loop–closure and velocity constraints.	rad, rad/s

2.1 Geometry and generalized coordinates

The mechanism has three rigid bars and four revolute joints. Bar 1 rotates about the ground point O, bar 3 rotates about the ground point C, and bar 2 connects the moving points A and B. The ground anchors are

$$O = [0.15,\ 0.00] \text{ m}, \qquad C = [0.00,\ 0.40] \text{ m},$$

so the ground offset is

$$\mathbf{r}_{OC} = \begin{bmatrix} -0.15 \\ 0.40 \end{bmatrix} \text{ m}.$$

Link lengths are

$$L_1 = 0.12 \text{ m}, \qquad L_2 = 0.50 \text{ m}, \qquad L_3 = 0.19 \text{ m}.$$

Angles are gathered as

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix},$$

where q_1 is the absolute angle of bar 1 about O; q_2 is the relative angle of bar 2 with respect to bar 1 at A; q_3 is the absolute angle of bar 3 about C. In the nominal assembled pose,

$$q_1 = 0, \qquad q_3 = 0.4899 \text{ rad } (\approx 28.07^\circ),$$

while q_2 is fixed by loop closure.

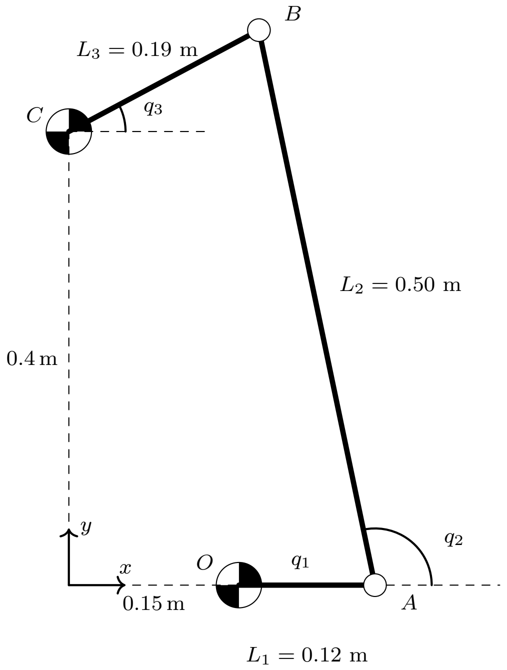


Figure 2.1: Geometry of the rigid four–bar.

Table 2.2: Fixed geometric and physical parameters used in the rigid four–bar model.

Symbol	**Description**	**Value**	**Units (SI)**
O	Lower ground revolute joint position in the global frame	(0.150, 0.000)	m
C	Upper ground revolute joint position in the global frame	(0.000, 0.400)	m
$\mathbf{r}_{OC}$	Ground offset vector from O to C	(−0.150, 0.400)	m
L_1	Length of bar 1	0.12	m
L_2	Length of bar 2	0.50	m
L_3	Length of bar 3	0.19	m
m_1	Mass of bar 1	1	kg
m_2	Mass of bar 2	1	kg
m_3	Mass of bar 3	1	kg

Continued on next page

Table 2.2: Fixed geometric and physical parameters used in the rigid four–bar model. (Continued)

Symbol	**Description**	**Value**	**Units (SI)**
I_1	Planar inertia of bar 1 about its centre of mass	0.0012	$\mathrm{kg\,m^2}$
I_2	Planar inertia of bar 2 about its centre of mass	0.020833	$\mathrm{kg\,m^2}$
I_3	Planar inertia of bar 3 about its centre of mass	0.003008	$\mathrm{kg\,m^2}$
g	Gravitational acceleration (pointing in negative y)	9.81	$\mathrm{m/s^2}$
Δt	Time step size for time integration (rigid four–bar tests)	0.01	s
ρ_∞	Target spectral radius at infinity (generalized–α scheme)	0	–

2.2 Kinematics, energies, and open–chain dynamics

Each bar $i \in \{1,2,3\}$ has mass m_i, length L_i, and planar inertia $I_i = \frac{1}{12} m_i L_i^2$. The centers of mass in global coordinates are

$$\mathbf{p}_{G1} = \begin{bmatrix} 0.15 + \frac{L_1}{2}\cos q_1 \\ \frac{L_1}{2}\sin q_1 \end{bmatrix}$$

$$\mathbf{p}_{G2} = \begin{bmatrix} 0.15 + L_1\cos q_1 + \frac{L_2}{2}\cos(q_1+q_2) \\ L_1\sin q_1 + \frac{L_2}{2}\sin(q_1+q_2) \end{bmatrix}$$

$$\mathbf{p}_{G3} = \begin{bmatrix} \frac{L_3}{2}\cos q_3 \\ 0.40 + \frac{L_3}{2}\sin q_3 \end{bmatrix}.$$

Differentiating gives velocities and then the kinetic energy T. The gravitational potential U uses gravity g along $-y$. With the Lagrangian $L = T - U$, the *open–chain* equations of motion are

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{c}(\dot{\mathbf{q}}, \mathbf{q}) = \boldsymbol{\tau} \tag{2.1}$$

where $\mathbf{M}$ is the 3×3 mass matrix and $\mathbf{c}$ collects gravity and velocity–dependent terms. For the chosen coordinates:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} \frac{1}{3}m_1L_1^2 + m_2L_1^2 + \frac{1}{3}m_2L_2^2 + m_2L_1L_2\cos q_2 & \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2\cos q_2 & 0 \\ \frac{1}{3}m_2L_2^2 + \frac{1}{2}m_2L_1L_2\cos q_2 & \frac{1}{3}m_2L_2^2 & 0 \\ 0 & 0 & \frac{1}{3}m_3L_3^2 \end{bmatrix}$$

$$\mathbf{c}(\dot{\mathbf{q}}, \mathbf{q}) = \begin{bmatrix} -\Big(\dot{q}_1 + \frac{\dot{q}_2}{2}\Big)m_2L_1L_2\,\dot{q}_2\sin q_2 + m_1g\,\frac{L_1}{2}\cos q_1 + m_2g\Big(L_1\cos q_1 + \frac{L_2}{2}\cos(q_1+q_2)\Big) \\ \frac{1}{2}m_2L_1L_2\,\dot{q}_1^2\sin q_2 + m_2g\,\frac{L_2}{2}\cos(q_1+q_2) \\ \frac{m_3L_3g}{2}\cos q_3 \end{bmatrix}$$

2.3 Loop closure and constraint equations

The two moving tips A and B must meet the tip of bar 3, so two position constraints are enforced:

$$\boldsymbol{\phi}(\mathbf{q}) = \begin{bmatrix} 0.15 + L_1\cos q_1 + L_2\cos(q_1+q_2) - L_3\cos q_3 \\ L_1\sin q_1 + L_2\sin(q_1+q_2) - 0.40 - L_3\sin q_3 \end{bmatrix} = \mathbf{0}.$$

The constraint Jacobian (often denoted $\mathbf{J}_c$ in the literature) is here written as

$$\mathbf{K}(\mathbf{q}) = \begin{bmatrix} -L_1\sin q_1 - L_2\sin(q_1+q_2) & -L_2\sin(q_1+q_2) & -L_3\sin q_3 \\ L_1\cos q_1 + L_2\cos(q_1+q_2) & L_2\cos(q_1+q_2) & L_3\cos q_3 \end{bmatrix}.$$

Time–differentiating gives velocity and acceleration compatibility:

$$\mathbf{K}(\mathbf{q})\,\dot{\mathbf{q}} = \mathbf{0}, \qquad \mathbf{K}(\mathbf{q})\,\ddot{\mathbf{q}} = \mathbf{k}(\dot{\mathbf{q}}, \mathbf{q}) = -\,\dot{\mathbf{K}}(\dot{\mathbf{q}}, \mathbf{q})\,\dot{\mathbf{q}}. \tag{2.2}$$

These relations support both approaches below.

2.4 Implicit constrained dynamics (KKT, index–3 DAE)

Idea

The implicit way adds Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^2$ that represent the unknown joint reactions. The equations form a KKT system:

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{K}(\mathbf{q})^{\mathsf{T}} \\ \mathbf{K}(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\,\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \mathbf{c}(\dot{\mathbf{q}}, \mathbf{q}) \\ \mathbf{k}(\dot{\mathbf{q}}, \mathbf{q}) \end{bmatrix}, \qquad \boldsymbol{\phi}(\mathbf{q}) = \mathbf{0}. \tag{2.3}$$

This is an index–3 DAE: positions, velocities, and accelerations must all be consistent with the constraints [9, 10].

Time stepping

Time integration uses the generalized–α method [6]. Two parameters (α_m, α_f) (chosen from a target spectral radius ρ_∞) control how much high–frequency content is damped while keeping second–order accuracy. At each step:

1. Predictor: build provisional $(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1})$.
2. Newton loop: assemble residuals

$$\mathbf{r}_q = \mathbf{M}\,\ddot{\mathbf{q}} - (\boldsymbol{\tau} - \mathbf{c}) + \mathbf{K}^{\mathsf{T}}\boldsymbol{\lambda}, \qquad \mathbf{r}_\lambda = \boldsymbol{\phi}(\mathbf{q}) \tag{2.4}$$

   and the effective tangent

$$\mathbf{S} = \begin{bmatrix} \beta_p\mathbf{M} + \gamma_p\mathbf{C}_t + \mathbf{K}_t & \mathbf{K}^{\mathsf{T}} \\ \mathbf{K} & \mathbf{0} \end{bmatrix}, \quad \beta_p = \frac{1-\alpha_m}{\beta(1-\alpha_f)\Delta t^2}, \quad \gamma_p = \frac{\gamma}{\beta(1-\alpha_f)\Delta t}.$$

   Here $\mathbf{K}_t = \partial\mathbf{H}/\partial\mathbf{q}$ and $\mathbf{C}_t = \partial\mathbf{H}/\partial\dot{\mathbf{q}}$ are consistent linearizations of the generalized forces.

3. Update: solve for corrections to $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda})$ until the combined residual norm is below tolerance.

Pros and cons

Pros: constraints are enforced at position level (no drift); no need to choose an independent coordinate; works robustly across most configurations. Cons: each step solves a coupled linear system of size $n + m$; the DAE needs consistent initial data and a well–conditioned $\mathbf{K}$; with $\rho_\infty < 1$ the method adds controlled high–frequency damping, which slightly reduces amplitudes in conservative tests unless steps and tolerances are very tight.

```
% Residuals
rq   = M(q)*qdd - (params.tau - c(q,qd)) + K(q).’*lam;
rlam = phi(q);

% Effective tangent
S11 = params.betap*M(q) + params.gammap*Ct(q,qd) + Kt(q,qd);
S12 = K(q).’;
S21 = K(q);
S22 = zeros(2);

% Solve [S11 S12; S21 S22]*[dq; dlam] = -[rq; rlam]
```

Listing 2.1: Implicit generalized-α step (KKT DAE).

2.5 Explicit geometric reduction (single ODE)

Idea

The explicit way picks one angle as the independent coordinate,

$$y = q_1,$$

and solves the two position constraints for the other two angles. This gives two geometric *branches*:

$$\boldsymbol{\gamma}_j(y) = \begin{bmatrix} y \\ q_2^{(j)}(y) \\ q_3^{(j)}(y) \end{bmatrix}, \qquad j \in \{1,2\}.$$

Differentiating gives the reduction Jacobian and the kinematic bias,

$$\mathbf{G}(y) = \frac{\partial \boldsymbol{\gamma}_j}{\partial y}, \qquad \mathbf{g}(y, \dot{y}) = \dot{\mathbf{G}}(y)\, \dot{y}. \tag{2.5}$$

Replacing $\mathbf{q} = \boldsymbol{\gamma}_j(y)$ and $\dot{\mathbf{q}} = \mathbf{G}(y)\dot{y}$ in the open–chain equations produces the scalar ODE

$$M_y(y)\, \ddot{y} + c_y(\dot{y}, y) = \tau_y(y) \tag{2.6}$$

with

$$M_y = \mathbf{G}^\mathsf{T}\mathbf{M}\,\mathbf{G}, \qquad c_y = \mathbf{G}^\mathsf{T}\left(\mathbf{c} + \mathbf{M}\,\mathbf{g}\right), \qquad \tau_y = \mathbf{G}^\mathsf{T}\boldsymbol{\tau}.$$

The ODE is integrated with an adaptive explicit Runge–Kutta method (order 4/5).

Pros and cons

Pros: the state dimension reduces to one; evaluation is fast; if the correct branch is kept and tolerances are tight, the motion matches the reference very closely. Cons: one must choose and keep the proper branch; near dead–center poses the mapping becomes ill–conditioned; explicit RK is not symplectic and can show slow energy drift for large steps.

```
% State y = [q1; dq1]
q   = gamma(y(1));          % [q1; q2; q3]
G   = Gmap(y(1));           % dgamma/dy (3x1)
gk  = Gdot(y(1), y(2));     % bias term dG/dt * ydot

My   = G.'*M(q)*G;
qy   = [y(2); G(2)*y(2); G(3)*y(2)];
cy   = G.'*( c(q, qy) + M(q)*gk );
tauy = G.'*params.tau;
```

```

dydt = [ y(2); (tauy - cy)/My ];
```

Listing 2.2: Explicit reduction: right-hand side for the scalar ODE.

2.6 Initial conditions and consistency

Starting values must satisfy the constraints. Given (q_1, q_3) and the geometry, q_2 comes from $\boldsymbol{\phi}(\mathbf{q}_0) = \mathbf{0}$ by choosing the branch that matches the build. Velocities must satisfy $\mathbf{K}(\mathbf{q}_0)\,\dot{\mathbf{q}}_0 = \mathbf{0}$. In the implicit formulation, initial accelerations and multipliers come from the KKT system with the selected torques and gravity. If these consistency conditions are not met, the DAE may show large transient constraint forces and spurious oscillations [9, 10].

2.7 Numerical behavior and comparison

The explicit geometric reduction replicates the commercial multibody reference to plotting accuracy over the entire test window with matched physics and numerically comparable tolerances. This agreement validates the reduced mapping $\gamma_j(y)$ and its Jacobian $G(y)$, as well as the accuracy of the symbolic expressions for the mass matrix and generalized forces. In other words, the single-DOF formulation captures the dynamic evolution of the rigid four-bar with virtually no discernible loss of fidelity once the proper geometric branch is chosen.

In contrast, when integrated with $\rho_\infty < 1$ and moderate tolerances, the implicit DAE formulation shows a mild amplitude decay. This behavior is anticipated: the trajectory is repeatedly projected back onto the constraint manifold by the Newton iterations, and the generalized-α scheme introduces controlled high-frequency numerical dissipation. Over longer horizons, the combination of these effects results in a slight phase shift and a small but discernible decrease in oscillation amplitude. These discrepancies are not structural; they diminish when the time step is decreased, the nonlinear tolerances are tightened, and the algorithmic damping is eliminated by setting $\rho_\infty \to 1$. The two formulations converge toward identical trajectories in the limit of extremely small steps and extremely stringent tolerances.

In general, the two strategies work well together. As long as the mechanism stays on a clearly defined geometric branch and away from singular poses, the explicit route is very effective and numerically light; in these situations, it provides near-reference accuracy with little overhead. In contrast, the implicit route is more reliable and systematic throughout the whole configuration space. It can handle transitions, near-singular configurations, and arbitrary load cases without requiring special handling because it enforces the loop closure directly at the algebraic level. Because of these factors, the implicit DAE functions as a trustworthy baseline and verification tool across the wider configuration range, while the explicit formulation is well suited for quick simulation and control design once a branch is fixed.

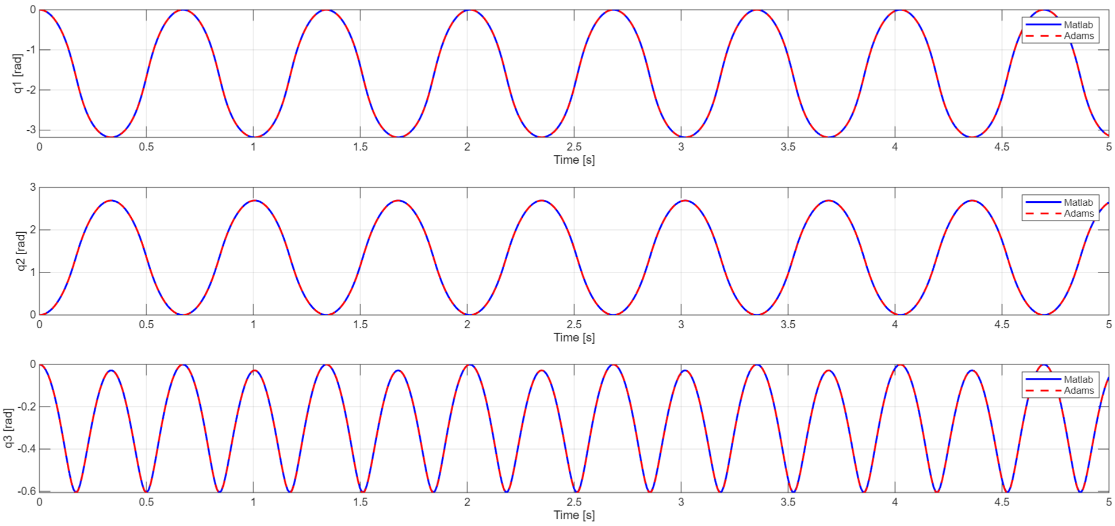


Figure 2.2: Joint angles: explicit ODE (solid blue) vs. commercial reference (dashed red).

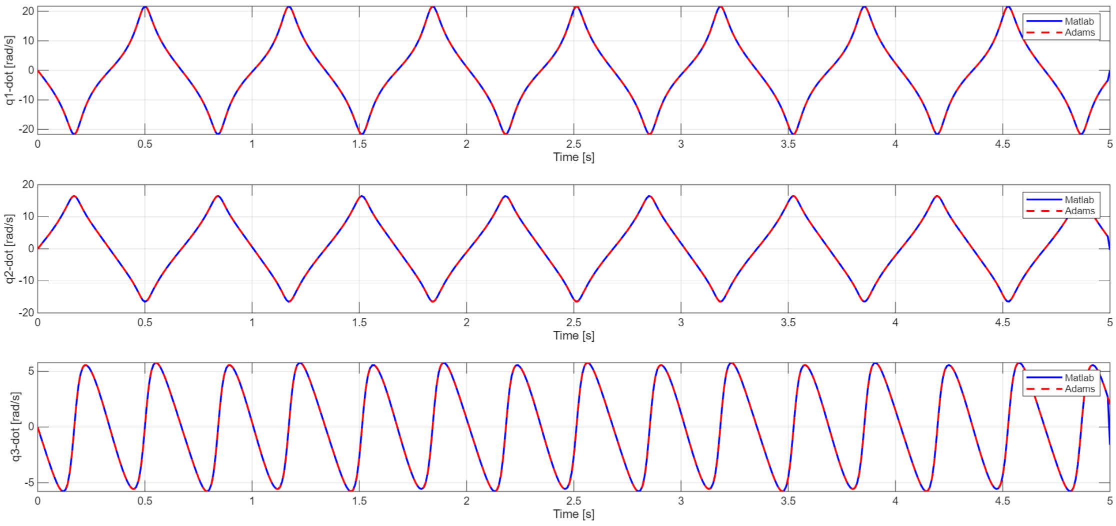


Figure 2.3: Angular velocities: explicit ODE (solid blue) vs. commercial reference (dashed red).

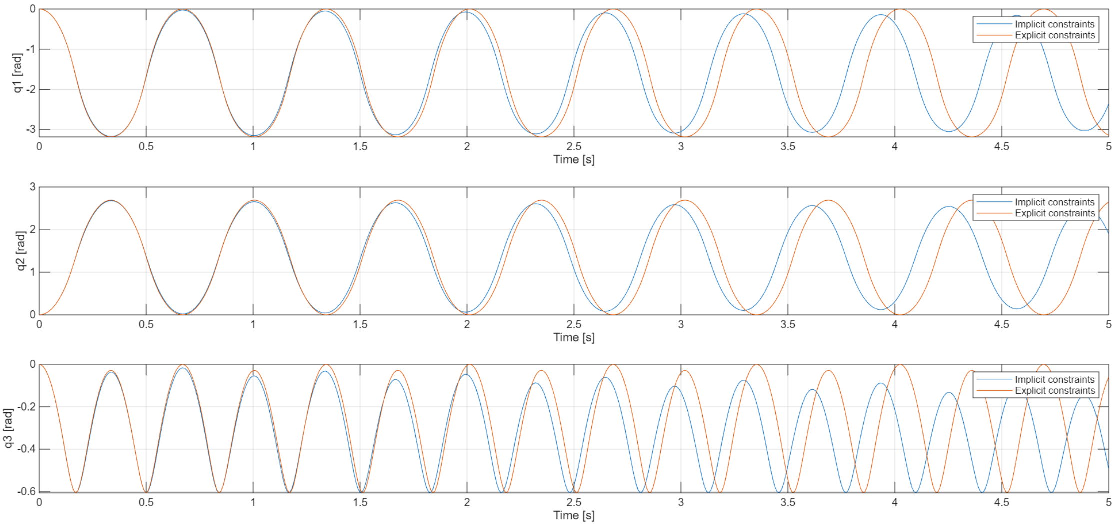


Figure 2.4: Joint angles: implicit DAE (blue) vs. explicit ODE (red). The implicit solution shows the expected amplitude reduction due to high–frequency damping.

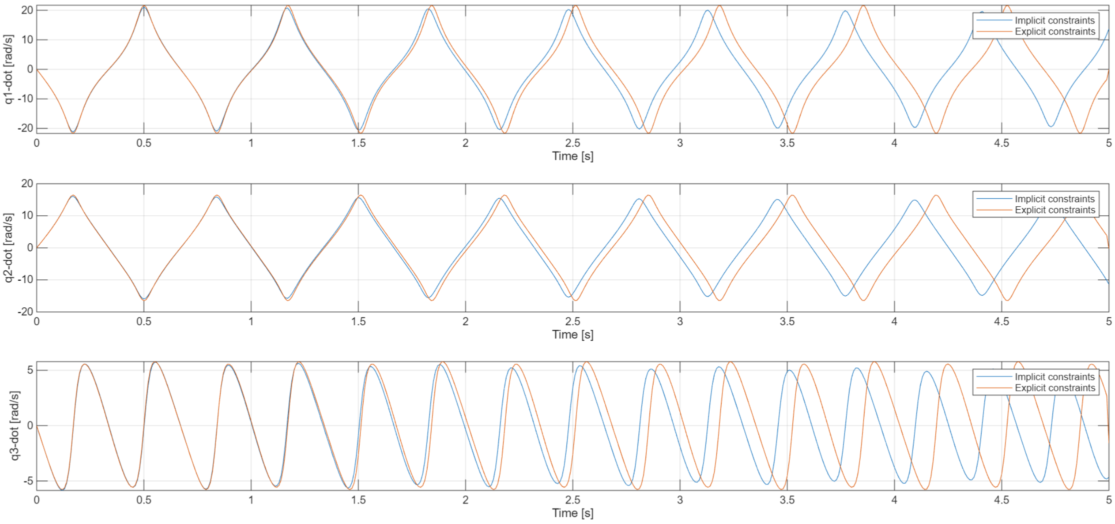
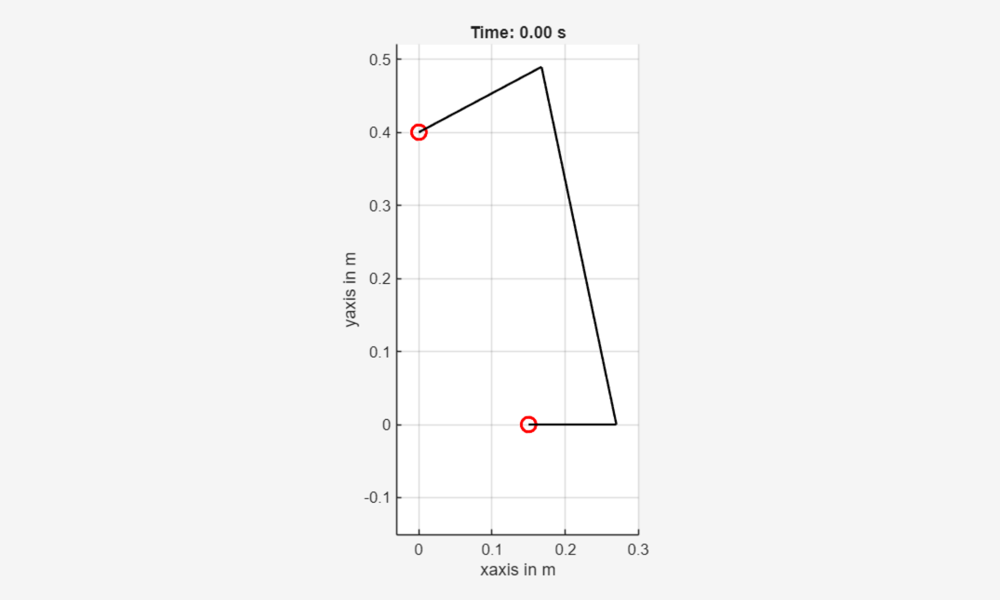
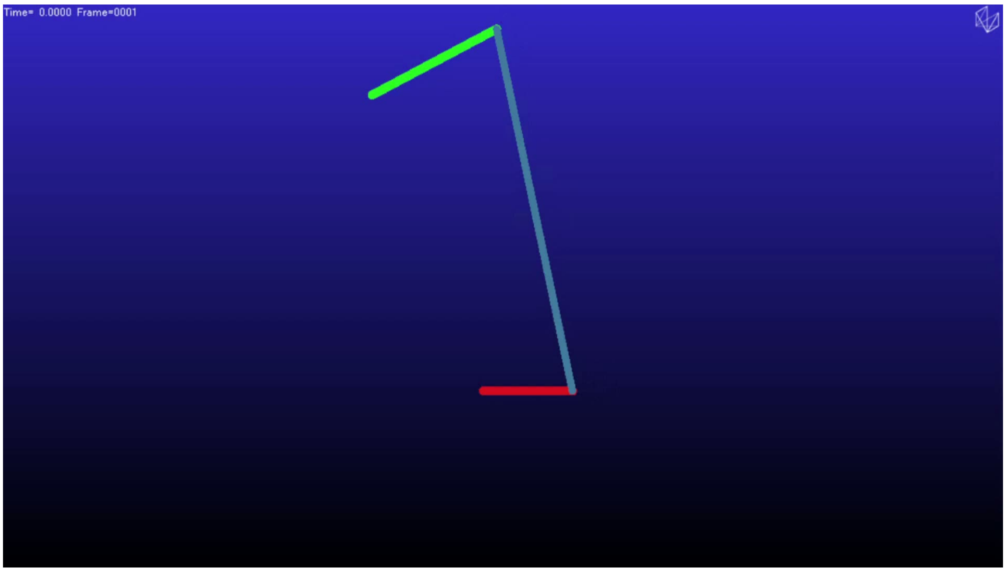


Figure 2.5: Angular velocities: implicit DAE (blue) vs. explicit ODE (red). The damping trend is consistent across all joints.



|◁ < ◁ ▷ > ▷| − ▸◂ +

Figure 2.6: Animation of MATLAB simulation with explicit solver (only gravity force applied)



|◁ < ◁ ▷ > ▷| − ▸◂ +

Figure 2.7: Animation of ADAMS simulation (only gravity force applied)

Chapter 3

Timoshenko Beam

This chapter documents the single Timoshenko beam module used for symbolic checks, analytical comparisons, static/modal validation, and as a flexible component to be coupled at leg level in the PACOMA architecture. The beam model is linear elastic (small strains/rotations), includes shear deformation and rotary inertia, and is discretized via two-node Timoshenko elements suitable for dynamics and control-oriented reduction [5, 11, 12, 13, 14, 15, 16, 17, 6, 18].

3.1 Kinematics and Strains

Let x denote the beam axis and y the transverse direction. Timoshenko kinematics (planar bending) assumes

$$u_x(x,y,t) = u(x,t) - y\,\omega_z(x,t), \qquad u_y(x,y,t) = v(x,t), \tag{3.1}$$

with nonzero strains

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} - y\,\frac{\partial \omega_z}{\partial x}, \qquad \gamma_{xy} = \frac{\partial v}{\partial x} - \omega_z. \tag{3.2}$$

Stress resultants (κ is the shear factor) are

$$N = EA\,u', \qquad M = EI\,\omega_z', \qquad V = \kappa GA\left(v' - \omega_z\right), \tag{3.3}$$

and yield the virtual internal work

$$\delta W_{\text{int}} = \int_0^L \Big[EA\,u'\delta u' + \kappa GA\,(v' - \omega_z)(\delta v' - \delta\omega_z) + EI\,\omega_z'\delta\omega_z'\Big]\,dx. \tag{3.4}$$

The kinetic energy is $T = \frac{1}{2}\int_0^L [\rho A(\dot{u}^2 + \dot{v}^2) + \rho I \dot{\omega}_z^2]\,dx$.

Table 3.1: Symbols, units, and conventions used in the Timoshenko beam chapter.

Symbol	**Description**	**Units (SI)**
x, y	Local coordinates: beam axis x, transverse direction y	m
X, Y	Global Cartesian coordinates of the beam axis	m
$u(x,t), v(x,t)$	Midline displacements: axial u, transverse v	m
$\omega_z(x,t)$	Cross-section rotation about the out-of-plane z-axis	rad
$u_x(x,y,t)$ $u_y(x,y,t)$	Components of the displacement field at (x,y)	m
ε_{xx}	Normal (axial) strain	–
γ_{xy}	Shear strain	–
κ_b	Bending curvature $\kappa_b = \omega_z'$	1/m
N, V, M	Internal resultants: axial force N, shear V, bending moment M	N, N, $\text{N} \cdot \text{m}$
$p_x(x,t)$	Axial distributed load per unit length	N/m
$q(x,t)$	Transverse distributed load per unit length	N/m
$m(x,t)$	Distributed bending moment per unit length (if present)	N
$N_0, N(x)$	Axial pre-load: constant N_0 or position-dependent $N(x)$	N
E	Young's modulus	Pa
ν	Poisson's ratio	–
G	Shear modulus $\big(= E/[2(1+\nu)]\big)$	Pa
κ	Shear correction factor (Timoshenko)	–
A	Cross-sectional area	m^2
I	Second moment of area for bending in the $x{-}y$ plane	m^4
b, h	Cross-section dimensions (width b, height h)	m
ρ	Material density	kg/m^3
ρA	Mass per unit length (translational inertia)	kg/m
ρI	Rotary inertia per unit length (about z)	$\text{kg} \cdot \text{m}$
L	Total beam length	m

Continued on next page

Table 3.1: Symbols, units, and conventions used in the Timoshenko beam chapter. (Continued)

Symbol	**Description**	**Units (SI)**
L_e	Length of a single finite element	m
N_e	Number of Timoshenko elements in the mesh	–
ξ	Reference (local) coordinate on [−1,1] for each element	–
$L_1(\xi), L_2(\xi)$	Linear Lagrange shape functions on $\xi \in [-1,1]$	–
α	Element orientation angle with respect to the global X-axis	rad
$\boldsymbol{q}_e$	Element nodal DOF vector $[u_1, v_1, \omega_1, u_2, v_2, \omega_2]^T$	– (m, rad)
$\boldsymbol{u}_e(x)$	Element displacement vector field, interpolated from $\boldsymbol{q}_e$	m, rad
$\boldsymbol{N}(\xi)$	Block shape matrix that maps $\boldsymbol{q}_e$ to (u, v, ω_z)	–
$\boldsymbol{B}(\xi)$	Strain–displacement matrix such that $\boldsymbol{\varepsilon}_e = \boldsymbol{B}\,\boldsymbol{q}_e$	1/m
$\boldsymbol{L}$	Differential operator used to build $\boldsymbol{B}(\xi)$	–
$\boldsymbol{\varepsilon}_e$	Element strain vector $[\varepsilon_{xx}, \gamma_{xy}, \kappa_b]^T$	–, –, 1/m
$\boldsymbol{\sigma}_e$	Element stress-resultant vector $[N, V, M]^T$	N, N, N · m
$\boldsymbol{D}$	Constitutive matrix diag$(EA, \kappa GA, EI)$	N, N, N · m
$\boldsymbol{M}_0$	Diagonal matrix diag$(\rho A, \rho A, \rho I)$ for kinetic energy	kg/m, kg/m, kg · m
$\boldsymbol{K}_e, \boldsymbol{M}_e$	Element stiffness and mass matrices in local axes	N/m, kg
$\boldsymbol{R}_e$	Local-to-global rotation matrix for one node	–
$\boldsymbol{T}_e$	Transformation matrix blkdiag$(\boldsymbol{R}_e, \boldsymbol{R}_e)$ for the element	–
$\boldsymbol{C}_e$	Boolean connectivity matrix assembling element DOFs into global DOFs	–
$\boldsymbol{K}_g(N_0)$	Geometric stiffness matrix associated with axial pre-load N_0	N/m
$\boldsymbol{K}, \boldsymbol{M}$	Global stiffness and mass matrices after assembly	N/m, kg

Continued on next page

Table 3.1: Symbols, units, and conventions used in the Timoshenko beam chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\boldsymbol{D}$	Global damping matrix (Rayleigh form $\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$)	$\mathrm{N \cdot s/m}$
$\boldsymbol{q}(t)$	Global vector of generalized coordinates (nodal DOFs)	– (m, rad)
$\dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$	Generalized velocity and acceleration vectors	m/s, $\mathrm{m/s^2}$
$\boldsymbol{F}(t)$	Global vector of generalized external forces	N, $\mathrm{N \cdot m}$
α, β	Rayleigh damping coefficients (mass- and stiffness-proportional)	$\mathrm{s^{-1}}$, s
ζ_i	Modal damping ratio of mode i	–
λ_i	Eigenvalue of the generalized eigenproblem $\boldsymbol{K}\boldsymbol{\phi}_i = \lambda_i \boldsymbol{M}\boldsymbol{\phi}_i$	$\mathrm{rad^2/s^2}$
ω_i	Natural circular frequency of mode i, $\omega_i = \sqrt{\lambda_i}$	rad/s
f_i	Natural frequency of mode i, $f_i = \omega_i/(2\pi)$	Hz
$\boldsymbol{\phi}_i$	Mode shape (eigenvector) associated with ω_i	–
ρ_∞	Target spectral radius at infinite frequency (generalized-α)	–
Δt	Time step of the time integration scheme	s
$(\cdot)'$	Derivative w.r.t. x (e.g. $u' = \partial u/\partial x$)	–
$\dot{(\cdot)}$	Derivative w.r.t. t (e.g. $\dot{v} = \partial v/\partial t$)	–

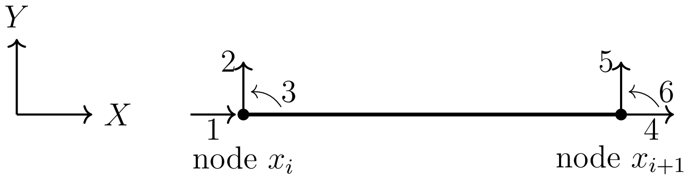


Figure 3.1: Two-node Timoshenko element and its nodal degrees of freedom $[u, v, \omega_z]$.

For a rectangular cross-section, the shear correction factor is usually taken as $\kappa = 5/6$. A more refined expression proposed by Cowper is $\kappa = \dfrac{10(1+\nu)}{12+11\nu}$, which accounts for non-uniform shear stress distribution [12]. For reference, the geometric properties of a rectangular section are $A = b\,h$ and $I = \dfrac{b\,h^3}{12}$, where h is the dimension along the transverse direction y and b is the width in the out-of-plane direction. Regarding the kinetic energy terms, the translational and rotary contributions appear as $\rho A(\dot{u}^2 + \dot{v}^2)$ and $\rho I\,\dot{\omega}_z^2$, respectively.

3.2 Strong Form and Boundary Conditions

Under the Timoshenko kinematics of Sec. 3.1, the beam is governed by a second–order system in the axial displacement $u(x,t)$, the transverse displacement $v(x,t)$, and the cross–section rotation $\omega_z(x,t)$. In the absence of an initial axial pre-stress, the strong form reads [13, 15]

$$\rho A\,\ddot{u} - \Big(EA\,u'\Big)' = p_x(x,t), \tag{3.5a}$$

$$\rho A\,\ddot{v} - \Big(\kappa GA\Big(v' - \omega_z\Big)\Big)' = q(x,t), \tag{3.5b}$$

$$\rho I\,\ddot{\omega}_z - \Big(EI\,\omega_z'\Big)' + \kappa GA\Big(v' - \omega_z\Big) = m(x,t), \tag{3.5c}$$

where $p_x(x,t)$ is an axial distributed load (if any), $q(x,t)$ is a transverse distributed load, and $m(x,t)$ is a distributed bending couple per unit length. All symbols are explained in Table 3.1.

Axial pre-load and geometric stiffness

If a constant axial force N_0 acts along the beam, the bending equation acquires the usual geometric-stiffness term, so that (3.5b) becomes

$$\rho A\,\ddot{v} - \Big(\kappa GA(v' - \omega_z)\Big)' + N_0\,v'' \;=\; q(x,t) \tag{3.6}$$

and analogously a spatially varying $N(x)$ induces a position-dependent geometric term. In the notation often used in structural dynamics, that contribution is represented at the discrete level by a geometric stiffness matrix $\boldsymbol{K}_g(N_0)$ added to the elastic $\boldsymbol{K}$ (see Sec. 3.3).

Boundary data (essential vs natural)

At the end sections $x = 0$ and $x = L$, boundary conditions can be specified either as essential (Dirichlet) conditions on the primary fields,

$$u = \bar{u}, \qquad v = \bar{v}, \qquad \omega_z = \bar{\omega},$$

or as natural (Neumann) conditions on the resultants

$$N = EA\,u', \qquad V = \kappa GA\,(v' - \omega_z), \qquad M = EI\,\omega_z'. \tag{3.7}$$

Typical cases are:

- Clamped end: $u = 0,\ v = 0,\ \omega_z = 0$.
- Simply supported: $v = 0,\ M = 0$ (axial u free or prescribed depending on the problem).
- Free end: $N = 0,\ V = \bar{P}$ (or $V = 0$), $M = \bar{M}$ (or $M = 0$).

Connection to the FEM discretization

The goal of the finite element method is to approximate (3.5) in space by interpolating the unknowns with shape functions. Over each two–node Timoshenko element e of length L_e,

$$u(x) \approx \boldsymbol{N}(x)\,\boldsymbol{u}_e, \qquad v(x) \approx \boldsymbol{N}(x)\,\boldsymbol{v}_e, \qquad \omega_z(x) \approx \boldsymbol{N}(x)\,\boldsymbol{\omega}_e,$$

with $\boldsymbol{N} = [L_1, L_2]$ the linear shape functions in $\xi \in [-1{,}1]$ and the element nodal vector $\boldsymbol{q}_e = [u_1,\, v_1,\, \omega_1,\, u_2,\, v_2,\, \omega_2]^T$. The weak form obtained from (3.5a)–(3.5c) leads to the element matrices reported in Sec. 3.3; assembling them over the mesh yields the semi-discrete model $\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{F}(t)$ used for the simulations and analytical checks.

3.3 Weak Form and Two-Node Timoshenko Element

Variational (weak) form.

Let $\delta u, \delta v, \delta\omega_z$ be admissible virtual fields. By multiplying the strong form of Sec. 3.2 by the virtual fields and integrating by parts, the weak statement is

$$\int_0^L \Big[EA\,u'\,\delta u' \;+\; \kappa GA\,\Big(v' - \omega_z\Big)\Big(\delta v' - \delta\omega_z\Big) \;+\; EI\,\omega_z'\,\delta\omega_z' \Big]\,dx \;=$$

$$\int_0^L \Big(p_x\,\delta u + q\,\delta v + m\,\delta\omega_z\Big)\,dx \;+\; \Big[N\,\delta u + V\,\delta v + M\,\delta\omega_z \Big]_{x=0}^{x=L} \tag{3.8}$$

with resultants $N = EA\,u'$, $V = \kappa GA\,(v' - \omega_z)$, $M = EI\,\omega_z'$. Essential (Dirichlet) boundary conditions set the corresponding natural (Neumann) terms to zero at the constrained ends.

Two-node interpolation (one element).

Discretize the beam into N_e Timoshenko elements. On a generic two-node element e of length L_e, use linear C^0 interpolation for the three primary fields:

$$u(x) \approx \sum_{i=1}^{2} L_i(\xi)\, u_i, \qquad v(x) \approx \sum_{i=1}^{2} L_i(\xi)\, v_i, \qquad \omega_z(x) \approx \sum_{i=1}^{2} L_i(\xi)\, \omega_i,$$

with reference coordinate $\xi \in [-1,1]$ and Lagrange polynomials $L_1 = \frac{1-\xi}{2},\ L_2 = \frac{1+\xi}{2}$. Collect nodal DOFs (Degrees of Freedom) into

$$\boldsymbol{q}_e = [\, u_1,\, v_1,\, \omega_1,\, u_2,\, v_2,\, \omega_2 \,]^T, \qquad \boldsymbol{u}_e(x) = \boldsymbol{N}(\xi)\, \boldsymbol{q}_e,$$

where the block shape matrix is

$$\boldsymbol{N}(\xi) = \begin{pmatrix} L_1 & 0 & 0 & L_2 & 0 & 0 \\ 0 & L_1 & 0 & 0 & L_2 & 0 \\ 0 & 0 & L_1 & 0 & 0 & L_2 \end{pmatrix}.$$

Reference mapping and derivatives.

For an arbitrarily oriented straight element connecting (X_1, Y_1) to (X_2, Y_2),

$$X(\xi) = L_1 X_1 + L_2 X_2, \qquad Y(\xi) = L_1 Y_1 + L_2 Y_2,$$

the Jacobian is $\det J = \frac{L_e}{2}$ with $L_e = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$. Chain rule gives

$$\frac{dL_i}{dx} = \frac{dL_i}{d\xi}\frac{d\xi}{dx} = \frac{2}{L_e}\frac{dL_i}{d\xi}, \qquad \frac{dL_1}{d\xi} = -\tfrac{1}{2}, \quad \frac{dL_2}{d\xi} = +\tfrac{1}{2}. \tag{3.9}$$

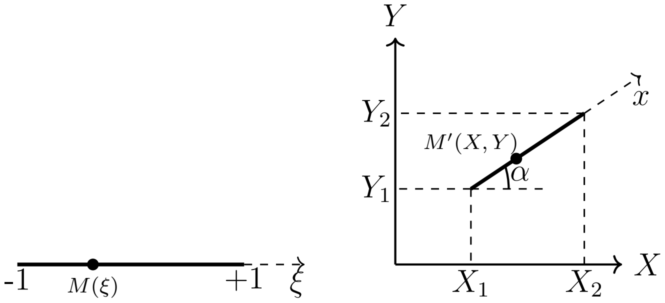
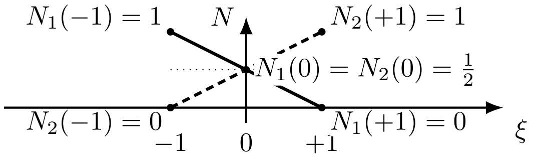


Figure 3.2: Local coordinate system (left) – global Cartesian coordinate system (right)



$N_1(\xi) = \dfrac{1-\xi}{2}, \quad N_2(\xi) = \dfrac{1+\xi}{2}$

Shape matrix rows:

$\boldsymbol{N}_1 = [N_1,0,0,N_2,0,0]$ (axial u)
$\boldsymbol{N}_2 = [0,N_1,0,0,N_2,0]$ (transverse v)
$\boldsymbol{N}_3 = [0,0,N_1,0,0,N_2]$ (rotation θ)

Figure 3.3: Linear two-node shape functions on $\xi \in [-1,1]$.

Discrete kinematics and strain–displacement operator.

Define the element "strain" vector $\boldsymbol{\varepsilon}_e = [\,\varepsilon_{xx},\ \gamma_{xy},\ \kappa_b\,]^T = [\,u',\ (v' - \omega_z),\ \omega_z'\,]^T$, with bending curvature $\kappa_b = \omega_z'$. Introduce the differential operator

$$\boldsymbol{L}^T = \begin{pmatrix} \dfrac{d}{dx} & 0 & 0 \\ 0 & \dfrac{d}{dx} & -1 \\ 0 & 0 & \dfrac{d}{dx} \end{pmatrix}, \qquad \Rightarrow \quad \boldsymbol{\varepsilon}_e = \boldsymbol{B}(\xi)\,\boldsymbol{q}_e, \quad \boldsymbol{B}(\xi) = \boldsymbol{L}^T\boldsymbol{N}(\xi).$$

For the linear two-node element this yields the explicit matrices

$$\boldsymbol{B}(\xi) = \begin{pmatrix} -\dfrac{1}{L_e} & 0 & 0 & \dfrac{1}{L_e} & 0 & 0 \\ 0 & -\dfrac{1}{L_e} & -L_1(\xi) & 0 & \dfrac{1}{L_e} & -L_2(\xi) \\ 0 & 0 & -\dfrac{1}{L_e} & 0 & 0 & \dfrac{1}{L_e} \end{pmatrix}.$$

Constitutive law (compact form).

In local element axes the stress–strain relation is

$$\boldsymbol{\sigma}_e = \begin{bmatrix} N \\ V \\ M \end{bmatrix} = \underbrace{\operatorname{diag}\Big(EA,\ \kappa GA,\ EI\Big)}_{\boldsymbol{D}}\,\boldsymbol{\varepsilon}_e,$$

which is the matrix form of $\sigma = \boldsymbol{D}\,\varepsilon$ used in your scalar expression.

Element matrices (local axes).

Using the weak form, Gauss quadrature on $\xi \in [-1{,}1]$, and $\det J = L_e/2$, the stiffness and mass matrices are

$$\boldsymbol{K}_e = \int_{-1}^{1} \boldsymbol{B}^T(\xi)\, \boldsymbol{D}\, \boldsymbol{B}(\xi)\, \frac{L_e}{2}\, d\xi, \tag{3.10}$$

$$\boldsymbol{M}_e = \int_{-1}^{1} \boldsymbol{N}^T(\xi)\; \underbrace{\operatorname{diag}\Big(\rho A,\ \rho A,\ \rho I\Big)}_{\boldsymbol{M}_0}\; \boldsymbol{N}(\xi)\, \frac{L_e}{2}\, d\xi. \tag{3.11}$$

(Equivalently, one may write $\boldsymbol{M}_e$ using separate $\boldsymbol{N}_u, \boldsymbol{N}_v, \boldsymbol{N}_\omega$ selectors.)

Local-to-global rotation and assembly.

If the element makes an angle α with the global X-axis, use

$$\boldsymbol{R}_e = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{T}_e = \operatorname{blkdiag}(\boldsymbol{R}_e, \boldsymbol{R}_e),$$

so that $\boldsymbol{K}_e^{(g)} = \boldsymbol{T}_e^T \boldsymbol{K}_e \boldsymbol{T}_e$ and $\boldsymbol{M}_e^{(g)} = \boldsymbol{T}_e^T \boldsymbol{M}_e \boldsymbol{T}_e$. With the Boolean connectivity $\boldsymbol{C}_e$,

$$\boldsymbol{K} = \sum_{e=1}^{N_e} \boldsymbol{C}_e^T\, \boldsymbol{K}_e^{(g)}\, \boldsymbol{C}_e, \qquad \boldsymbol{M} = \sum_{e=1}^{N_e} \boldsymbol{C}_e^T\, \boldsymbol{M}_e^{(g)}\, \boldsymbol{C}_e.$$

Application case (cantilever with tip load).

For the standard test used throughout: left end clamped ($u = v = \omega_z = 0$ at $x = 0$); right end free with a transverse tip load P (and/or tip moment M) applied via a consistent nodal force vector. After imposing boundary conditions, the semi-discrete dynamics reads

$$\boldsymbol{M}\, \ddot{\boldsymbol{q}} + \boldsymbol{D}\, \dot{\boldsymbol{q}} + \boldsymbol{K}\, \boldsymbol{q} = \boldsymbol{F}(t), \qquad \boldsymbol{D} = \alpha \boldsymbol{M} + \beta \boldsymbol{K} \tag{3.12}$$

with (α, β) selected as in Sec. 3.7.

Selective Reduced Integration (anti-locking)

For slender elements ($L_e/h \gg 1$) the shear part of (3.10) may over-stiffen the response (shear locking). An effective remedy is *Selective Reduced Integration* (SRI): evaluate the κGA contribution (the shear block) with one Gauss point, while keeping full integration (two Gauss points for linear mapping) for the axial EA and bending EI terms [13, 15, 16]. This restores the correct shear compliance without degrading axial/bending accuracy.

3.4 Loads, Damping, and the Semi-Discrete Equations

The external actions considered here act per unit length along the beam axis. A transverse distributed load $q(x,t)$ produces a shear field and, after spatial discretization, a consistent vector of generalized forces obtained by weighting q with the transverse shape functions of each element and integrating over its span. The special case of self-weight is directly included as $q = \rho A g$, so that the gravity load is treated on the same footing as any other distributed excitation.
After assembling all element contributions and enforcing essential boundary conditions, the beam reduces to a finite number of generalized coordinates collected in the vector $\boldsymbol{q}(t)$. The dynamics is governed by the second-order, linear, time-varying system

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}}(t) \;+\; \boldsymbol{D}\,\dot{\boldsymbol{q}}(t) \;+\; \boldsymbol{K}\,\boldsymbol{q}(t) \;=\; \boldsymbol{F}(t), \tag{3.13}$$

where:

- $\boldsymbol{M}$ is the (symmetric, positive definite) mass matrix accounting for translational and rotary inertia of the Timoshenko kinematics;
- $\boldsymbol{K}$ is the (symmetric, positive semidefinite) elastic stiffness that includes axial, shear, and bending contributions; geometric stiffness terms can be added if an axial pre-load is present;
- $\boldsymbol{D}$ represents structural dissipation. Here a standard and effective description is Rayleigh damping, which assumes a linear combination of inertia- and stiffness-proportional contributions: $\boldsymbol{D} = \alpha\,\boldsymbol{M} + \beta\,\boldsymbol{K}$ [13, 16, 17].

The coefficients α and β are not arbitrary: they are chosen so that two selected modes of the structure exhibit prescribed damping ratios. This is particularly convenient for vibration problems, where one wishes to control dissipation in the low-frequency band without over-damping the higher modes (see Sec. 3.7 for the identification formulas and their interpretation).

Equation (3.13) expresses conservation of linear momentum in generalized coordinates. The left-hand side collects inertia, dissipation, and elastic restoring forces; the right-hand side $\boldsymbol{F}(t)$ collects all generalized excitations (distributed loads mapped to nodal form, tip forces/moments, base motions, etc.). Its structure guarantees existence and uniqueness of solutions once initial conditions $\boldsymbol{q}(0)$, $\dot{\boldsymbol{q}}(0)$ are specified.

3.5 Time Integration: Newmark and Generalized-α

The semi-discrete system (3.13) is advanced in time with single-step schemes that approximate accelerations and velocities over a finite time step Δt. Two closely related families are used in practice.

3.5.1 Newmark's family.

Newmark introduces two parameters β and γ that regulate how acceleration is interpolated within the time step [18]. With $\gamma = \frac{1}{2}$ and $\beta \in [1/6,\ 1/4]$ the method is second-order accurate and unconditionally stable for linear systems. Intuitively, $\gamma = \frac{1}{2}$ centers the evaluation of acceleration in the interval $[t_n, t_{n+1}]$, while β controls the implicitness of the displacement update. Larger β values increase numerical damping of very high frequencies while preserving accuracy in the frequency band of interest.

3.5.2 Generalized-α scheme.

The generalized-α method preserves the same formal accuracy of Newmark but introduces a tunable parameter $\rho_\infty \in [0{,}1]$, the desired spectral radius at infinite frequency. Choosing $\rho_\infty < 1$ gently damps spurious high-frequency content (which is routinely generated by discretization and non-smooth loadings) while leaving the low-frequency response essentially unchanged. The method automatically sets its internal parameters $(\alpha_m, \alpha_f, \beta, \gamma)$ as functions of ρ_∞ (cf. [6]); the resulting linear system has a fixed, *effective* tangent per step that combines $\boldsymbol{M}$, $\boldsymbol{D}$ and $\boldsymbol{K}$. In practice, $\rho_\infty \in [0.5,\ 0.9]$ provides robust transients without polluting the band where the physical modes lie.

Interpretation. Both schemes can be viewed as energy-consistent quadratures in time. Newmark (with $\beta = \frac{1}{4}, \gamma = \frac{1}{2}$) is often used as a baseline; generalized-α adds a controlled "filter" at the tail of the spectrum to suppress numerical chatter without affecting the dynamics of interest.

3.6 Modal Analysis and Worked Examples

3.6.1 The generalized eigenvalue problem.

In the absence of external forces and damping, the free vibration of the beam is governed by

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}}(t) + \boldsymbol{K}\,\boldsymbol{q}(t) = \boldsymbol{0}. \tag{3.14}$$

Looking for harmonic solutions $\boldsymbol{q}(t) = \boldsymbol{\phi}\, e^{i\omega t}$ leads to the generalized eigenproblem

$$\boldsymbol{K}\,\boldsymbol{\phi} = \lambda\,\boldsymbol{M}\,\boldsymbol{\phi}, \qquad \lambda = \omega^2. \tag{3.15}$$

Each eigenpair $(\lambda_i, \boldsymbol{\phi}_i)$ defines a *natural circular frequency* $\omega_i = \sqrt{\lambda_i}$ and a *mode shape* $\boldsymbol{\phi}_i$, i.e. a deformation pattern that oscillates sinusoidally at frequency $f_i = \omega_i/(2\pi)$ without change of shape.

3.6.2 Orthogonality and normalization.

Because $\boldsymbol{M}$ and $\boldsymbol{K}$ are symmetric, eigenvectors associated with distinct eigenvalues satisfy the classical orthogonality relations

$$\boldsymbol{\phi}_i^T \boldsymbol{M}\,\boldsymbol{\phi}_j = \delta_{ij}, \qquad \boldsymbol{\phi}_i^T \boldsymbol{K}\,\boldsymbol{\phi}_j = \lambda_i\,\delta_{ij}.$$

The first identity is a choice of *mass normalization*: it fixes the arbitrary scale of each mode and is convenient for interpreting modal energies and effective masses.

3.7 Rayleigh Damping Identification

Rayleigh damping assigns to each mode i a damping ratio $\zeta_i = \frac{1}{2}(\alpha/\omega_i + \beta\,\omega_i)$. Selecting two target modes (ω_1, ω_2) and prescribing the same ratio ζ in both leads to two linear equations in the unknowns (α, β), whose solution is

$$\alpha = \frac{2\zeta\,\omega_1\omega_2}{\omega_1 + \omega_2}, \qquad \beta = \frac{2\zeta}{\omega_1 + \omega_2}.$$

The α-term dominates at low frequency (mass-proportional damping), whereas the β-term dominates at high frequency (stiffness-proportional damping). Placing the identification at ω_1 and ω_2 anchors dissipation in the band of interest; modes far below ω_1 and far above ω_2 become progressively under- and over-damped, respectively—an effect often desirable to regularize the numerical tail.

3.8 Validation with a Cantilever Beam Test Case

To validate the numerical model implemented in MATLAB and MSC Adams, a test case of modal analysis for a cantilever beam is considered. The reference case is taken from the validation section of the OnScale platform, which provides numerical results obtained from the software Ansys [19]. The cantilever beam has the following properties:

- Material: Steel

- Young's modulus: $E = 480\,GPa$
- Poisson's ratio: $\nu = 1 \cdot 10^{-6}$
- Density: $\rho = 7700\,kg/m^3$
- Length: $L = 1\,m$
- Cross-section: $b \times h = 0.04\,m \times 0.04\,m$

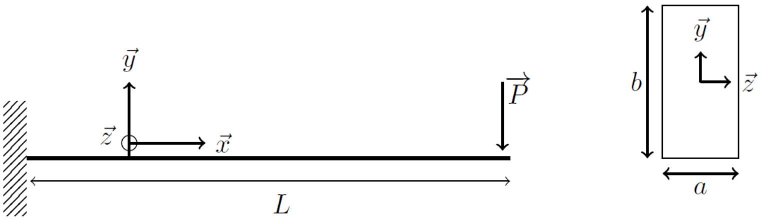


Figure 3.4: Timoshenko beam geometry and boundary conditions used throughout.

The boundary conditions impose a fixed constraint on one end of the beam, while the opposite end remains free. The natural frequencies of the beam are computed using the MATLAB code, the commercial software MSC Adams, and compared with the reference values provided by Ansys.
The natural frequencies were calculated in MATLAB by solving the generalized eigenvalue problem:

$$\boldsymbol{K}\boldsymbol{q} = \lambda \boldsymbol{M}\boldsymbol{q} \tag{3.16}$$

where λ represents the eigenvalues, and $\boldsymbol{q}$ the corresponding eigenvectors. The eigenvalues were extracted using the built-in `eig` function in MATLAB. The circular frequencies were then obtained by:

$$\omega_i = \sqrt{\lambda_i} \tag{3.17}$$

and the corresponding modal frequencies were computed as:

$$f_i = \frac{\omega_i}{2\pi} \tag{3.18}$$

The first three natural frequencies obtained are summarized in Table 3.2:

Mode	MATLAB [Hz]	MSC Adams [Hz]	Ansys [Hz]
1	51.1	50.97	50.97
2	322.4	317.7	317.5
3	913.3	881.97	880.6

Table 3.2: Comparison of the first three natural frequencies for the test-case cantilever beam

As shown in Table 3.2, the numerical results obtained from the MATLAB implementation are in excellent agreement with both the MSC Adams simulation and the reference solution provided by Ansys. The small differences observed are mainly due to numerical approximations and different implementations of the finite element formulation.
This validation test case confirms the reliability of the numerical model for predicting the natural frequencies of dynamic simulations of Timoshenko beams using the Finite Element Method. However, further analysis is required to validate the damping properties and the final displacement values.

3.9 Numerical results

This section focuses on the numerical results of the Timoshenko beam described in Figure 3.4 but with the geometrical and material properties described in Table 3.3.

Table 3.3: Fixed geometric, material, and numerical parameters used in the Timoshenko beam study.

Symbol	**Description**	**Value**	**Units (SI)**
L	Total beam length	0.15	m
b	Beam width (out-of-plane direction)	$2.5 \cdot 10^{-3}$	m
h	Beam height (in x–y plane)	$5 \cdot 10^{-3}$	m
E	Young's modulus of the beam material	$70{\cdot}10^{9}$	Pa
ν	Poisson's ratio	0.33	–
G	Shear modulus $G = E/[2(1+\nu)]$	$2.63{\cdot}10^{10}$	Pa
κ	Shear correction factor (Timoshenko)	5/6	–
ρ	Material density	2600	kg/m^3

Continued on next page

Table 3.3: Fixed geometric, material, and numerical parameters used in the Timoshenko beam study. (Continued)

Symbol	**Description**	**Value**	**Units (SI)**
N_e	Number of Timoshenko elements in the mesh	10	–
Δt	Time step of the time integration scheme	$1 \cdot 10^{-4}$	s
T_{end}	Final simulation time in the transient analysis	5	s
P	Vertical external force applied at the beam free-end	5	N
g	Acceleration of gravity	9.81	m/s^2
q	Distributed load due to gravity per unit length $q = \rho A g$	0.318	N/m

Comparing the outcomes from the MATLAB implementation and the MSC Adams software. Two different cases are analyzed: with and without gravity applied to the beam.

For both cases, the displacement of the beam on the right side in a static state can be calculated using the analytical formulation. Let ΔL represent the displacement of the free end under an applied force P:

$$\Delta L = \frac{PL^3}{3EI} \tag{3.19}$$

The corresponding displacement obtained from MATLAB and MSC Adams is compared to the analytical solution, showing excellent agreement with a small discrepancy, which is due to numerical approximations and to the discretization (i.e., the number of elements) used in the numerical analysis, affecting the precision of the results.

When gravity is applied, the beam is subjected to both the applied force and the self-weight distributed along its length. The static deflection is recalculated by including the gravity effect, which modifies the displacement equation to:

$$\Delta L_{gravity} = \frac{PL^3}{3EI} + \frac{qL^4}{8EI} \tag{3.20}$$

where $q = \rho A g$ is the distributed load per unit length due to gravity. The numerical results again show consistent agreement between MATLAB, MSC Adams, and the analytical solution, confirming the correctness of the implemented model in both cases.

The final displacements for both cases are summarized in Table 3.4, when a force of 5 N is applied.

Case	MATLAB [mm]	MSC Adams [mm]	Analytical [mm]
Without Gravity	3.0807	3.0875	3.0857
With Gravity	3.0918	3.0986	3.0968

Table 3.4: Comparison of final static displacements with and without gravity

Mode	MATLAB [Hz]	MSC Adams [Hz]
1	186.5	186.1
2	1177.6	1160.9
3	3337.8	3227.4
4	6659.7	6261.9
5	8656.8	8647.9

Table 3.5: Modal frequencies of the beam

The final static deflection is nearly the same in all three methods (analytical solution, MATLAB finite element model, and MSC Adams simulation), both with and without gravity, according to the results shown in Table 3.4. The slight variations in boundary condition implementation, solver tolerances, and numerical discretization (finite element mesh resolution) are all responsible for the small differences, which are on the order of a few microns. The physical properties of the beam are consistent with the limited impact of gravity on the total displacement: the self-weight contribution is significantly less than the displacement caused by the applied tip force due to the beam's very low mass per unit length, small span, and modest cross-section dimensions.

The correspondence between MATLAB and MSC Adams is equally satisfactory from a dynamic perspective. With shear deformation, rotary inertia, and consistent mass representation, the implemented Timoshenko formulation captures the expected dynamic behavior of the structure, as demonstrated by the close agreement of the natural frequencies listed in Table 3.5. The mode-shape visualizations in Sec. 3.10, which show the distinctive bending and axial patterns predicted by classical beam theory, further support this alignment between the two computational models. Overall, the findings show that the MATLAB finite element implementation is dependable for both modal and static predictions, matching the behavior of a commercial multibody platform in every test that was carried out.

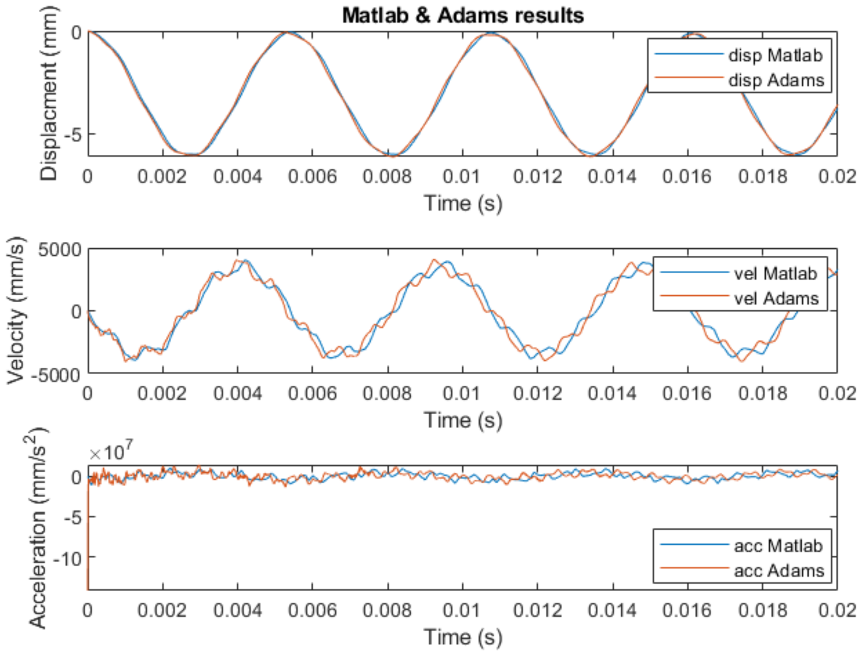


Figure 3.5: Displacement-Velocity-Acceleration of the last node (MATLAB and Adams results)

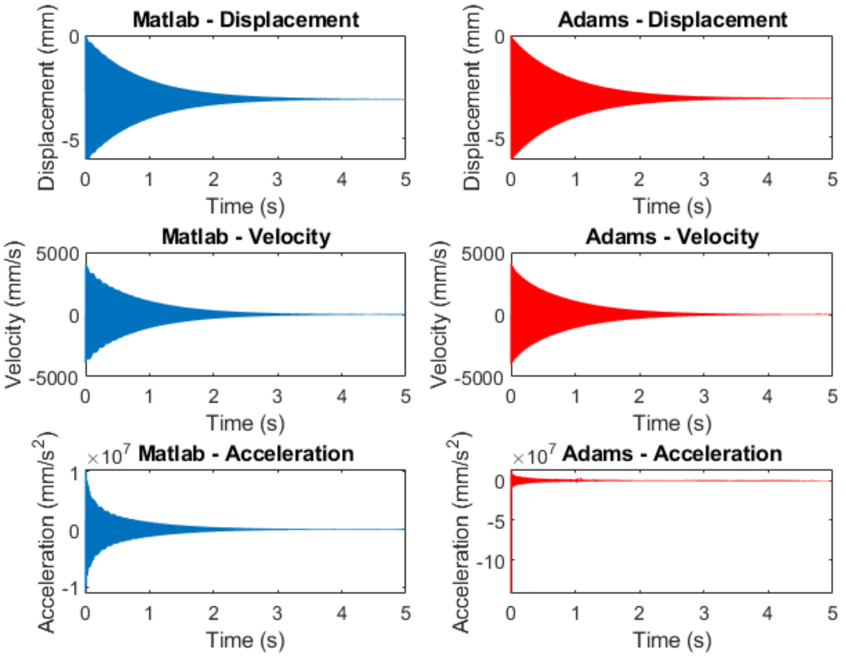


Figure 3.6: Comparison of damping results between MATLAB and MSC Adams

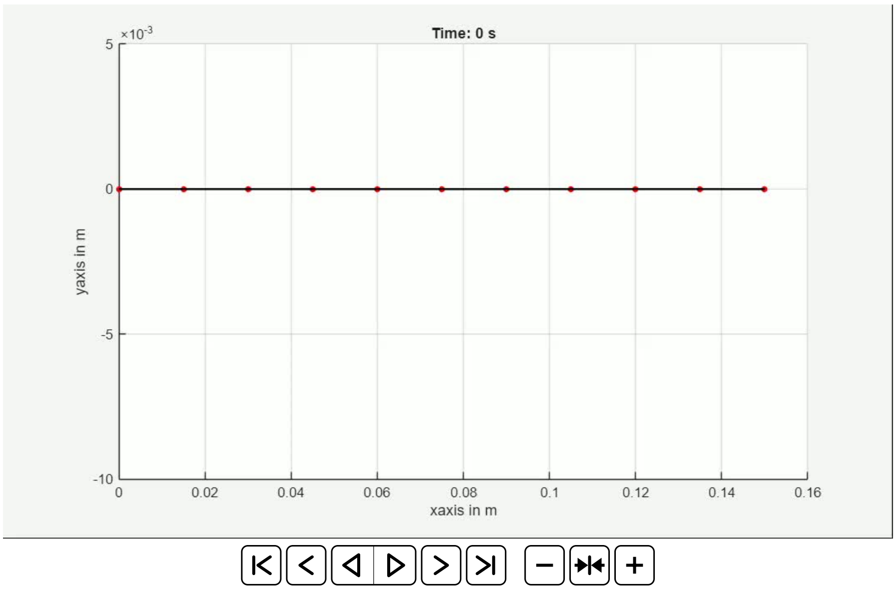


Figure 3.7: Animation of MATLAB simulation (only gravity force and vertical end-node force applied).

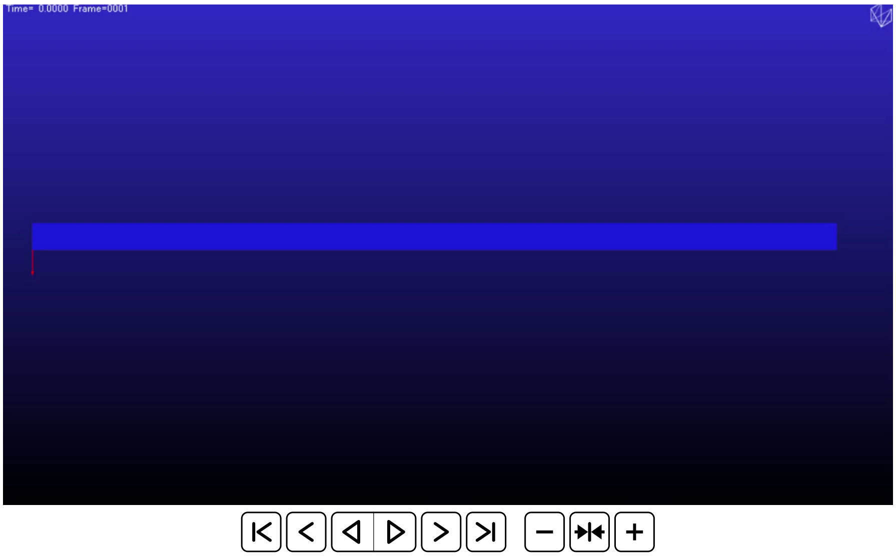


Figure 3.8: Animation of ADAMS simulation (only gravity force and vertical end-node force applied).

Figure 3.5 shows the displacement, the velocity, and the acceleration of the last node of the beam in the first instants of simulation. The results in Adams and MATLAB are very similar, showing the consistency of the MATLAB implementation code.

Figure 3.6 compares the results of the last node between MATLAB and Adams over a longer time period, highlighting the compatibility between the two simulations (with the same damping and same final values of displacement, velocity and acceleration).

This simulation generates a very high oscillation frequency of the beam, and due to the resulting computational load, only a few tenths of a second of the animation can be displayed in this document, as shown in Figures 3.7 and 3.8.

3.10 Natural frequencies

The following figures show the lowest five natural modes of the cantilever beam analyzed as shown in Table 3.5. In the figures, the color blue means low deformation (e.g. in all the figures the left end is blue because is clamped), while color the red color indicates large deformation.

Figure 3.9: Representation of the first frequency: 186.5 Hz

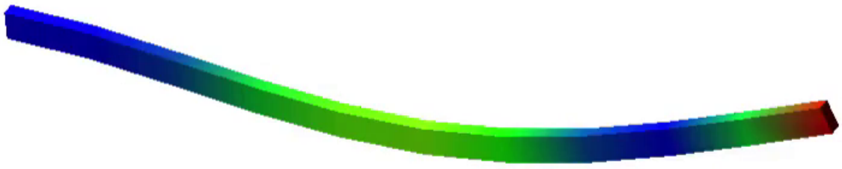


Figure 3.10: Representation of the second frequency: 1177.6 Hz

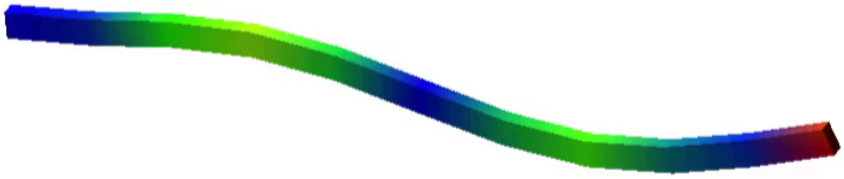


Figure 3.11: Representation of the third frequency: 3337.8 Hz

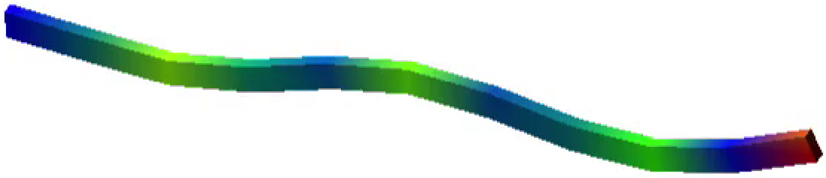


Figure 3.12: Representation of the fourth frequency: 6659.7 Hz

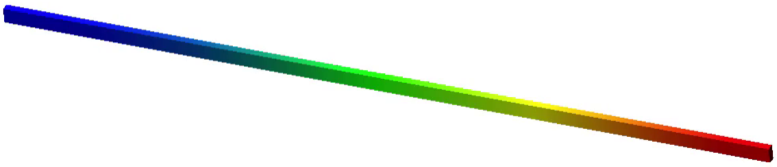


Figure 3.13: Representation of the fifth frequency: 8656.8 Hz (the beam lengthens and shortens linearly)

Chapter 4

Simplified Flexible Multi Body Simulation Model of PACOMA Leg Mechanism

This chapter describes a first rigid–flexible implementation of the four–bar mechanism, developed as an intermediate step between the purely rigid four–bar baseline and the final rigid–flexible formulation presented in the next chapter. The same physical system is considered: two rigid links of lengths L_1 and L_3 are grounded at their left ends and connected by a flexible coupler of length L_2, modeled as a planar Timoshenko beam. However, a number of modeling choices in this first attempt turned out to constrain the beam to behave almost as a rigid link, and to introduce unnecessary stiffness and numerical difficulties. These limitations motivated the revised formulation later adopted in the thesis.

The objective of this chapter is twofold: (i) to document the structure of the first rigid–flexible model (generalized coordinates, energies, constraints and solver), and (ii) to clearly identify the main inconsistencies with respect to the target physical problem and to the final implementation.

Table 4.1: Symbols used in the simplified flexible multi body simulation model chapter.

Symbol	**Description**	**Units (SI)**
L_1, L_2, L_3	Lengths of lower rigid link, flexible coupler, and upper rigid link	m

Continued on next page

Table 4.1: Symbols used in the simplified flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
L_0	Vertical offset between the two ground hinges (height of point C)	m
P_1	Ground hinge of the lower rigid link, $P_1 = (0.150{,}0)$ in global coordinates	m
q_1, q_2, q_3	Joint angles: absolute angle of link 1, relative angle between link 1 and the undeformed flexible link, absolute angle of link 3	rad
$\dot{q}_i, \ddot{q}_i$	First and second time derivatives of the joint angles	rad/s, rad/s^2
N_e	Number of Timoshenko beam elements used for the flexible link	–
N_n	Number of beam nodes, $N_n = N_e + 1$	–
b, h	Width and height of the rectangular cross-section of the flexible link	m
u_i, v_i, θ_i	Axial displacement, transverse displacement, and cross-section rotation at node i of the flexible link	m, m, rad
$\boldsymbol{q}$	Global vector of generalized coordinates $[q_1, q_2, q_3, q_{d,1}, \ldots, q_{d,n_\mathrm{d}}]^\mathsf{T}$	– (m, rad)
$\boldsymbol{q}_\mathrm{rig}$	Subvector of rigid coordinates, $(q_1, q_2, q_3)^\mathsf{T}$	rad
$\boldsymbol{q}_\mathrm{d}$	Subvector of active flexible DOFs of the beam	– (m, rad)
n_d	Number of active DOFs of the flexible link	–
m_1, m_3	Masses of the lower and upper rigid links	kg
ρ	Material density of the flexible link	kg/m^3
A	Cross-sectional area of the flexible link	m^2
I	Second moment of area of the flexible link	m^4
g	Gravitational acceleration	m/s^2
K_rig	Kinetic energy of the two rigid links	J
$U_\mathrm{grav,rig}$	Gravitational potential energy of the rigid links	J
$K_\mathrm{flex,trans}$	Translational kinetic energy of the flexible link	J

Continued on next page

Table 4.1: Symbols used in the simplified flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$K_{\text{flex,rot}}$	Rotational kinetic energy of the flexible link	J
$U_{\text{grav,flex}}$	Gravitational potential energy of the flexible link	J
U_{stress}	Elastic strain energy of the flexible link	J
$\mathcal{L}$	Lagrangian of the rigid–flexible system	J
$\mathbf{r}_e(x,t)$	Position vector of a representative point on element e in the global frame	m
x_e, x_{e+1}	Coordinates of the end points of element e along the beam axis	m
$\omega_e(x,t)$	Section rotation field along element e	rad
$y(x,t)$	Vertical coordinate of the deformed neutral axis	m
$\mathbf{K}$	Global stiffness matrix of the flexible link (assembled from element matrices)	N/m
$\mathbf{M}(\boldsymbol{q})$	Global mass matrix of the rigid–flexible system	kg
$\mathbf{h}(\boldsymbol{q},\dot{\boldsymbol{q}})$	Vector collecting Coriolis, centrifugal, gravitational and elastic generalized forces	N, $\text{N} \cdot \text{m}$
$\boldsymbol{\tau}$	Vector of applied generalized torques on the rigid links	$\text{N} \cdot \text{m}$
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers (constraint reactions)	N, $\text{N} \cdot \text{m}$
$\boldsymbol{\phi}(\boldsymbol{q})$	Loop–closure constraint vector	– (m)
$\mathbf{K}(\boldsymbol{q})$	Constraint Jacobian, $\partial\boldsymbol{\phi}/\partial\boldsymbol{q}$	–
$\mathbf{K}_{\text{rig}}(\boldsymbol{q})$	Part of the constraint Jacobian associated with the rigid coordinates	–
$\mathbf{r}_B^{(\text{first})}$	Tip position of the flexible link in the first model, computed as if the coupler were rigid	m
$\mathbf{S}_{\text{t}}$	Iteration (tangent) matrix of the generalized-α Newton scheme	–
$\mathbf{K}_{\text{t}}$	Tangent stiffness matrix of the internal forces	N/m
$\mathbf{C}_{\text{t}}$	Tangent damping matrix (non-proportional part)	$\text{N} \cdot \text{s/m}$

Continued on next page

Table 4.1: Symbols used in the simplified flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\beta_\mathrm{p}, \gamma_\mathrm{p}$	Effective generalized-α parameters multiplying $\mathbf{M}$ and $\mathbf{C}_\mathrm{t}$ in $\mathbf{S}_\mathrm{t}$	–
ρ_∞	Target spectral radius at infinite frequency for the generalized-α scheme	–
$\alpha_\mathrm{R}, \beta_\mathrm{R}$	Rayleigh damping coefficients (mass- and stiffness-proportional)	s^{-1}, s
$\mathbf{C}_\mathrm{R}^{(\mathrm{flex})}$	Rayleigh damping matrix acting on the flexible DOFs	$\mathrm{N} \cdot \mathrm{s/m}$
$\mathbf{M}_\mathrm{flex}$	Flexible-part submatrix of the mass matrix used in Rayleigh damping	kg
$\mathbf{K}_{t,\mathrm{flex}}$	Flexible-part submatrix of the tangent stiffness used in Rayleigh damping	N/m
$\mathbf{f}_\mathrm{damp}$	Damping generalized force vector used in the residual of the equations of motion	N, $\mathrm{N} \cdot \mathrm{m}$
$\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$	Generalized velocity and acceleration vectors	(m/s, rad/s), ($\mathrm{m/s^2}$, $\mathrm{rad/s^2}$)

4.1 Geometry, coordinates and finite element mesh

The geometry of the mechanism is the same used throughout the thesis. The lower rigid link of length L_1 is connected to the ground at the point

$$P_1 = (0.150,\ 0)\ \mathrm{m},$$

the upper rigid link of length L_3 is grounded at height L_0 along the vertical axis, and the flexible leaf spring of length L_2 connects the tip of link 1 to that of link 3. The configuration of the two rigid bars is described by the angles q_1 and q_3 measured from the horizontal, while q_2 denotes the relative rotation between the lower rigid link and the undeformed axis of the flexible link.

The flexible coupler is discretised as a 2D Timoshenko beam with N_e elements and $N_n = N_e + 1$ nodes. In this first implementation the number of elements is

$$N_e = 10,$$

and the flexible link is modeled as a rectangular beam with height $h = 1.2\,\mathrm{mm}$ and width $b = 77\,\mathrm{mm}$, matching the PACOMA prototype dimensions [1] and reported in Table 4.2. Each node has three mechanical degrees of freedom (DOFs): a longitudinal displacement u along the undeformed beam axis, a transverse displacement v, and a rotation θ of the cross–section. However, only the internal nodes $i = 2, \ldots, N_e$ are fully active. The first and last nodes are treated as pinned ends, with only the rotational DOF activated,

$$\text{node } 1: \quad (u_1, v_1, \theta_1) = (0{,}0, \theta_1), \qquad \text{node } N_n: \quad (u_{N_n}, v_{N_n}, \theta_{N_n}) = (0{,}0, \theta_{N_n}), \tag{4.1}$$

so that the translational DOFs at the beam ends are completely suppressed. Collecting the three rigid angles and the active nodal DOFs of the beam into a single vector, the generalized coordinates are written as

$$\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_{d,1} & \cdots & q_{d,n_\mathrm{d}} \end{bmatrix}^\mathsf{T} = \begin{bmatrix} \boldsymbol{q}_\mathrm{rig} \\ \boldsymbol{q}_\mathrm{d} \end{bmatrix}, \tag{4.2}$$

where $\boldsymbol{q}_\mathrm{rig} = (q_1, q_2, q_3)^\mathsf{T}$ and $\boldsymbol{q}_\mathrm{d}$ collects the n_d active DOFs of the flexible link, as defined by the mesh numbering.

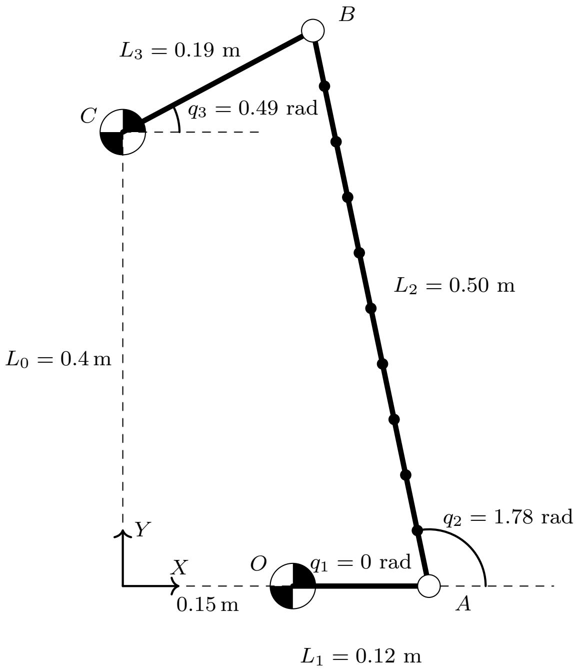


Figure 4.1: Rigid-flexible model's initial geometry.

Figure 4.1 shows the initial geometry of the rigid-flexible model. L_1 is the length of the lower rigid bar connected to the actuator. L_3 is the length of the upper rigid bar, representing the end effector. L_2 is the length of the titanium leaf spring before buckling due to the pretension applied by the tendon, as indicated in Table 1.1.

Table 4.2: Fixed geometric, material, and numerical parameters used in the simplified flexible multi body simulation model.

Symbol	**Description**	**Value**	**Units (SI)**
P_1	Ground hinge of the lower rigid link in global coordinates	(0.150, 0.000)	m
L_0	Vertical offset between the two ground hinges (height of point C)	0.40	m
L_1	Length of the lower rigid link	0.12	m
L_2	Length of the flexible coupler (beam)	0.50	m
L_3	Length of the upper rigid link	0.19	m
b	Width of the rectangular cross-section of the flexible link	$77 \cdot 10^{-3}$	m
h	Height (thickness) of the rectangular cross-section of the flexible link	$1.2 \cdot 10^{-3}$	m
N_e	Number of Timoshenko beam elements used for the flexible link	10	–
N_n	Number of beam nodes ($N_n = N_e + 1$)	11	–
E	Young's modulus of the flexible link material	$120 \cdot 10^{9}$	Pa
ν	Poisson's ratio of the flexible link material	35	–
G	Shear modulus of the flexible link material	$4.444 \cdot 10^{10}$	Pa
κ	Shear correction factor (Timoshenko) for the flexible link	5/6	–
ρ	Material density of the flexible link	4502	kg/m^3
m_1	Mass of the lower rigid link	1	kg
m_3	Mass of the upper rigid link	1	kg
g	Gravitational acceleration	9.81	m/s^2

Continued on next page

Table 4.2: Fixed geometric, material, and numerical parameters used in the simplified flexible multi body simulation model. (Continued)

Symbol	**Description**	**Value**	**Units (SI)**
Δt	Time step size used in the generalized–α integration	$1 \cdot 10^{-3}$	s
ρ_∞	Target spectral radius at infinity (generalized–α scheme)	0	–

4.2 Energies in the first rigid–flexible formulation

4.2.1 Rigid links

The two rigid bars are modelled as slender rods with lumped rotation around their left hinge. Their kinetic energy is

$$K_{\text{rig}} = \frac{1}{6} m_1 L_1^2 \dot{q}_1^2 + \frac{1}{6} m_3 L_3^2 \dot{q}_3^2, \tag{4.3}$$

and the gravitational potential reads

$$U_{\text{grav,rig}} = m_1 g \frac{L_1}{2} \sin q_1 + m_3 g \frac{L_3}{2} \sin q_3 + m_3 g L_0, \tag{4.4}$$

where the constant term $m_3 g L_0$ can be omitted without affecting the equations of motion.

4.2.2 Flexible link: kinetic energy

The flexible coupler is described in terms of nodal DOFs defined in the global frame. In this first implementation, the translational kinetic energy is evaluated by reconstructing the velocity of a representative point on each element using an analytical expression

$$\mathbf{r}_e(t) = \begin{bmatrix} x_e(t) \\ y_e(t) \end{bmatrix},$$

which depends on q_1, q_2 and on the element nodal DOFs through the shape functions. The squared velocity $\|\dot{\mathbf{r}}_e\|^2$ is obtained by symbolic differentiation and then expressed back in terms of the generalised velocities. The translational kinetic energy of the flexible link is finally computed as

$$K_{\text{flex,trans}} = \frac{1}{2} \rho A \sum_{e=1}^{N_e} \int_{x_e}^{x_{e+1}} \|\dot{\mathbf{r}}_e(x,t)\|^2 \, \mathrm{d}x, \tag{4.5}$$

which is numerically evaluated by a one–point Gauss rule per element in the actual implementation.

The rotational kinetic energy is constructed in a similar spirit. For each element, the section rotation is interpolated by the rotational shape functions and the squared angular velocity is integrated along the beam, leading to

$$K_{\text{flex,rot}} = \frac{1}{2}\,\rho I \sum_{e=1}^{N_e} \int_{x_e}^{x_{e+1}} \omega_e(x,t)^2 \,\mathrm{d}x. \tag{4.6}$$

Here, however, the rotation field is approximated as the sum of a “rigid” component proportional to $\dot{q}_1 + \dot{q}_2$ and a flexible contribution depending on the nodal rotations, which tends to underestimate the coupling between axial, shear and bending motions for larger deformations.

4.2.3 Flexible link: gravitational and elastic energy

The gravitational energy of the flexible link is defined by integrating the vertical position of the deformed neutral axis over the beam length. In this first model, the deformed geometry of the beam is reconstructed from the undeformed mesh `geom` and from the nodal displacements u and v, using the current orientation of each element, but still with respect to the initial straight configuration of the leaf. The resulting potential energy reads

$$U_{\text{grav,flex}} = \rho A g \sum_{e=1}^{N_e} \int_{x_e}^{x_{e+1}} y(x,t)\,\mathrm{d}x. \tag{4.7}$$

The strain energy U_{stress} is computed using a linear Timoshenko element stiffness matrix assembled over the initial geometry. For each element, a 6×6 stiffness matrix is obtained in a local frame aligned with the initial axis of the beam and then rotated to the global frame. The global stiffness matrix $\mathbf{K}$ is built by standard finite element assembly and the elastic energy is written as

$$U_{\text{stress}} = \frac{1}{2}\,\boldsymbol{q}_{\text{d}}^{\mathsf{T}}\,\mathbf{K}\,\boldsymbol{q}_{\text{d}}. \tag{4.8}$$

This choice is formally consistent with a small–deflection Timoshenko beam, but it neglects the change of orientation of each element and any geometric nonlinearity in the strain measure. An additional scalar parameter allows the stiffness energy to be scaled by a large factor, effectively turning the flexible link into a nearly rigid coupler for debugging purposes.

4.2.4 Lagrangian and equations of motion

Collecting all contributions, the Lagrangian of the first rigid–flexible model is

$$\mathcal{L} = K_{\text{rig}} + K_{\text{flex,trans}} + K_{\text{flex,rot}} - \Big(U_{\text{grav,rig}} + U_{\text{grav,flex}} + U_{\text{stress}}\Big). \tag{4.9}$$

The equations of motion are obtained by standard Lagrange equations with constraints. In compact form, the dynamics can be written as

$$\mathbf{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \mathbf{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} - \mathbf{K}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda}, \qquad \boldsymbol{\phi}(\boldsymbol{q}) = \mathbf{0}, \tag{4.10}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{h}$ collects Coriolis, centrifugal, gravitational and elastic terms, $\boldsymbol{\tau}$ is the vector of generalized torques applied to the rigid links, and $\boldsymbol{\lambda}$ are the Lagrange multipliers enforcing the loop–closure constraint $\boldsymbol{\phi}(\boldsymbol{q}) = \mathbf{0}$.

4.3 Rigid loop–closure constraint and its consequences

The most critical modelling choice in this first implementation lies in the treatment of the loop–closure constraint. The mechanism is topologically equivalent to a four–bar linkage and the constraint ensures that the tip of the flexible link coincides with the tip of the upper rigid link. In the first model, however, the constraint vector was written as if the coupler were rigid:

$$\boldsymbol{\phi}(\boldsymbol{q}) = \begin{bmatrix} 0.150 + L_1 \cos q_1 + L_2 \cos(q_1 + q_2) - L_3 \cos q_3 \\ L_1 \sin q_1 + L_2 \sin(q_1 + q_2) - L_3 \sin q_3 - L_0 \end{bmatrix}. \tag{4.11}$$

The position of point B, at the end of the flexible link, is thus expressed as

$$\mathbf{r}_B^{(\text{first})} = \begin{bmatrix} 0.150 + L_1 \cos q_1 + L_2 \cos(q_1 + q_2) \\ L_1 \sin q_1 + L_2 \sin(q_1 + q_2) \end{bmatrix}, \tag{4.12}$$

which coincides exactly with the position of a rigid link of length L_2 attached to link 1 with relative angle q_2. Crucially, the nodal displacements of the flexible link do not appear in (4.11). As a result, the Jacobian matrix of the constraints has the structure

$$\mathbf{K}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{q}} = \begin{bmatrix} \mathbf{K}_{\text{rig}}(\boldsymbol{q}) & \mathbf{0}_{2\times n_{\text{d}}} \end{bmatrix}, \tag{4.13}$$

so that the flexible DOFs are completely absent from the loop–closure conditions. This modelling choice has two important consequences:

- the kinematic closure of the mechanism is enforced as if the flexible link were a rigid bar of length L_2, driven only by (q_1, q_2, q_3);
- the elastic displacements of the beam cannot participate in satisfying the loop–closure; they only enter the internal elastic forces and energies, but they do not influence the position of point B in the constraint equations.

In practice, this means that the three rigid angles q_1, q_2 and q_3 evolve almost exactly as in the rigid four–bar baseline, while the flexible link is forced to deform "on top of" the rigid motion without any possibility to relieve the kinematic closure through its own DOFs. The last node of the beam is pinned in translation, and the loop constraint sees only the undeformed, straight configuration of the coupler. The combined effect is that the flexible link behaves as an extremely stiff element, with very small displacements at the tip and a dynamic response that is much closer to a rigid four–bar than to the leaf–spring mechanism observed in the multibody reference model.

4.4 Time integration and damping in the first DAE solver

4.4.1 Generalized-α scheme with fixed tangent

The system (4.10) is an index–3 differential–algebraic equation and is integrated in time using a generalized-α method. The solver advances the state $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and the Lagrange multipliers $\boldsymbol{\lambda}$ over each time step by solving a nonlinear system obtained from the discrete form of the equations of motion and constraints.

In the first implementation, the iteration matrix for the Newton scheme was assembled as

$$\mathbf{S}_\mathrm{t} = \begin{bmatrix} \beta_\mathrm{p}\,\mathbf{M} + \gamma_\mathrm{p}\,\mathbf{C}_\mathrm{t} + \mathbf{K}_\mathrm{t} & \mathbf{K}^\mathsf{T} \\ \mathbf{K} & \mathbf{0} \end{bmatrix}, \tag{4.14}$$

where $\mathbf{K}_\mathrm{t}$ is the tangent matrix of the internal forces with respect to $\boldsymbol{q}$, including the contribution of $\mathbf{K}^\mathsf{T}\boldsymbol{\lambda}$, and $\mathbf{C}_\mathrm{t}$ is the tangent of the non–proportional damping term. To reduce computational cost, the tangent stiffness matrix was evaluated once per time step, at the beginning of the Newton iteration, and then kept fixed ("frozen" tangent) until convergence.

While this strategy is acceptable for moderately nonlinear problems, it turned out to be fragile in combination with the stiff constraints discussed above. The frozen tangent worsens the conditioning of the iteration matrix, especially when the configuration changes appreciably within a single time step, and tends to increase the number of Newton iterations or even cause convergence issues for larger time steps.

4.4.2 Rayleigh damping and its limitations

The first solver also introduced Rayleigh damping on top of the tangent damping matrix, with coefficients α_R and β_R tuned on the first two bending modes of the flexible beam. In the implementation, a Rayleigh matrix is assembled at each time step by combining the mass and stiffness contributions on the flexible DOFs only,

$$\mathbf{C}_{\mathrm{R}}^{(\mathrm{flex})} = \alpha_{\mathrm{R}}\,\mathbf{M}_{\mathrm{flex}} + \beta_{\mathrm{R}}\,\mathbf{K}_{t,\mathrm{flex}}, \tag{4.15}$$

where $\mathbf{M}_{\mathrm{flex}}$ and $\mathbf{K}_{t,\mathrm{flex}}$ are the submatrices of the mass and tangent stiffness associated with the flexible coordinates. This Rayleigh contribution is added to the tangent damping matrix in the iteration matrix of the generalized-α scheme. At the residual level, however, the damping forces are evaluated using a simplified Rayleigh term. In practice, only a mass–proportional term of the form

$$\mathbf{f}_{\mathrm{damp}} \approx \alpha_{\mathrm{R}}\,\mathbf{M}\,\dot{\boldsymbol{q}} \tag{4.16}$$

is applied, while the stiffness–proportional part is neglected and the damping is effectively distributed over all DOFs, not only the flexible ones. As a result, the damping forces that enter the equations of motion are not fully consistent with the Rayleigh matrix used in the tangent operator of the Newton iterations.

Combined with the generalized–α parameter choice $\rho_\infty = 0$, which maximizes algorithmic damping of high–frequency components, this setting leads to a strongly damped response of the bending modes. In practice, the elastic vibrations of the leaf spring are heavily attenuated.

4.5 Observed behavior and mismatch with the target mechanism

The combination of a rigid loop–closure constraint, pinned translational DOFs at both ends of the beam and a very stiff linear elastic model produced a dynamic behavior that was qualitatively different from the expected one.

From a mechanical point of view, the flexible link could deform only in the interior nodes, without moving its tip relative to the rigid geometry imposed by (4.11). As a result, the curvature of the beam remained small and the vertical deflection of the tip was much smaller than in the multibody reference model of the leaf spring mechanism. The rigid angles q_1 and q_3 evolved almost identically to the rigid four–bar baseline, confirming that the flexible link was not allowed to redistribute the kinematic closure.

From a numerical point of view, the strongly constrained configuration and the frozen tangent occasionally led to poorly conditioned iteration matrices, requiring a large number of Newton iterations or very small time steps to achieve convergence. The overdamped nature of the time integrator further masked the dynamic contribution of the elastic modes.

These discrepancies highlighted that the first rigid–flexible formulation, although structurally sound from a finite element and DAE point of view, did not faithfully represent the mechanics of the actual leaf–spring mechanism and could not serve as a reliable basis for comparison with the multibody model.

4.6 Key differences with the final rigid–flexible formulation

The issues identified in this first implementation drove a substantial revision of the model, leading to the final rigid–flexible formulation presented in the next chapter. The main differences between the two versions can be summarised as follows:

- **Loop–closure constraint.** In the first model, the position of the coupler tip B in the loop constraint is computed as if the link were rigid and depends only on (q_1, q_2, q_3). In the final formulation, the position of point B is reconstructed from the current finite element geometry of the flexible link, using the same routine employed for the energies. As a consequence, the loop–closure constraint depends explicitly on the nodal DOFs and the elastic deformation contributes to satisfying the closure.
- **Boundary conditions of the flexible link.** The preliminary mesh activates only the rotational DOF at the first and last node of the beam, locking their translations. In the final model, the rightmost node of the flexible link has all three DOFs active, so that the tip position can move relative to the rigid geometry. This is essential to reproduce the measured tip motion and the Adams reference solution.
- **Co–rotational versus reference–configuration energies.** In the first implementation, the elastic stiffness matrix is assembled using element matrices referred to the initial straight configuration, and the coupling between rotations and translations in the kinetic and potential energies is only approximate. The final formulation consistently uses a co–rotational description: the current geometry of each element is computed from the generalized coordinates and the energies are evaluated in a local frame aligned with the deformed element. This improves objectivity and allows for large rotations of the leaf spring without spurious stiffening.
- **Treatment of the constraints in the solver.** The first solver introduces the loop–closure constraint directly into the Lagrangian via Lagrange multipliers and solves an index–3 DAE with the constraints enforced only at the acceleration level. In the final solver, the constraints are handled by a combination of Baumgarte stabilization and explicit projection of positions and velocities onto the constraint manifold at each time step, which significantly improves robustness and reduces drift.

- **Damping model.** Rayleigh damping is introduced in a simplified way in the first model and applied uniformly to all DOFs. In the final implementation, the Rayleigh coefficients are identified from the bending spectrum of the flexible link, and the associated damping is applied only to the flexible DOFs, while the generalized–α parameters are chosen to reduce numerical dissipation and preserve the physical damping of the system.

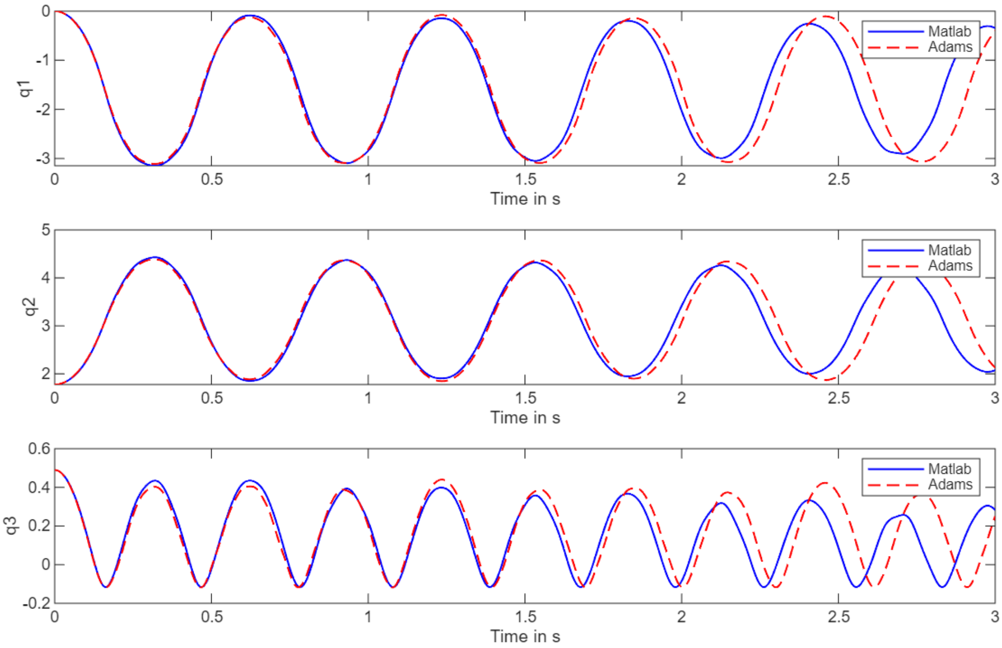


Figure 4.2: Trend of the angles of the two rigid links and the flexible beam (only gravity force applied)

In summary, the first rigid–flexible implementation provided a useful intermediate step to set up the finite element discretization, the symbolic generation of equations of motion and the DAE solver. However, the combination of a rigid loop–closure constraint, pinned boundary conditions at the beam tip and a linear stiffness matrix assembled in the reference configuration effectively constrained the flexible link to behave almost as a rigid coupler. For this reason, when the model is subject to the action of gravity alone, the deformations of the flexible nodes and the angle trends are similar between the MATLAB and Adams simulations, as can be seen in Figures 4.2 and 4.3 and in the animations shown in Figures 4.4 and 4.5. However, precisely because of this incorrect setting of the constraints, when attempting to flex the flexible beam by constraining the lower rigid rod and applying an external load to the upper rigid rod, the flexible beam does not flex and assume a completely wrong, even leading to model failure due to inconsistency between loop closure and internal stress.

The final formulation overcomes these limitations by consistently including the beam deformation in the loop closure, adopting a co–rotational description of the flexible link and revising the constraint enforcement and damping strategies.

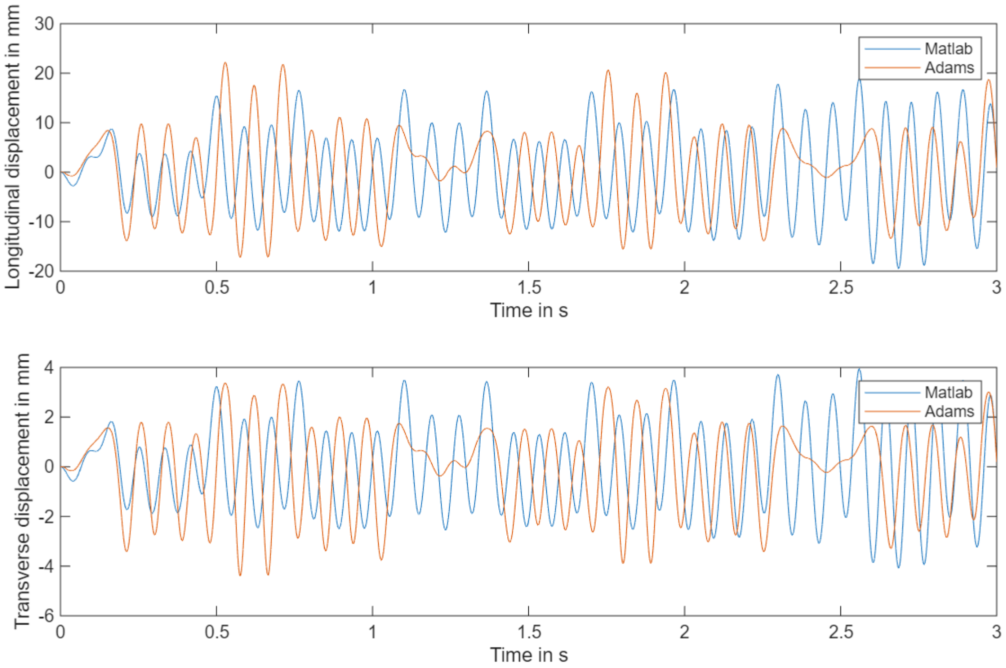


Figure 4.3: Longitudinal and transverse deformations of the middle node (only gravity force applied)

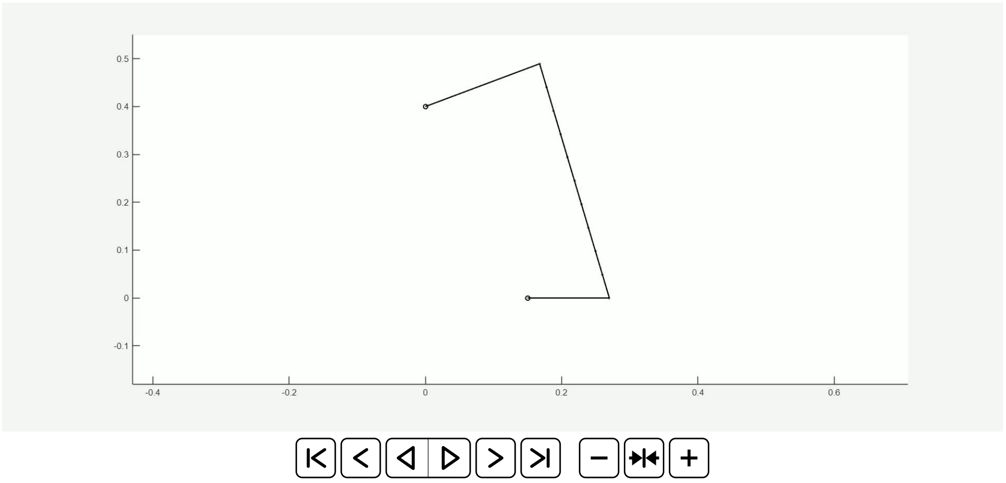


Figure 4.4: Animation of MATLAB simulation (only gravity force applied)

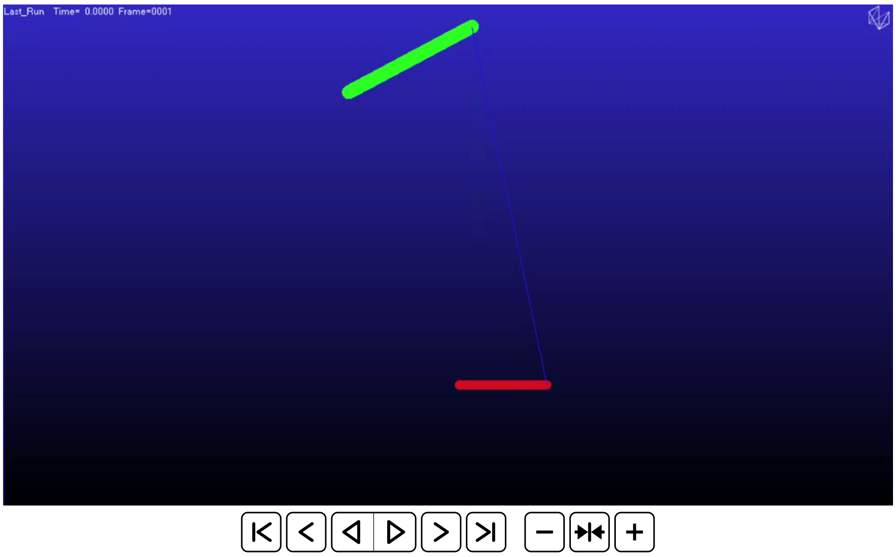


Figure 4.5: Animation of ADAMS simulation (only gravity force applied)

Chapter 5

Full Flexible Multi Body Simulation Model of PACOMA Leg Mechanism

5.1 Aim and modeling scope

This chapter develops a planar rigid–flexible model that represents a single leg of the parallel continuum mechanism in its operating configuration. The leg is composed of two rigid links and one flexible beam discretized by Timoshenko elements. The objective is to present the full derivation of the differential–algebraic equations, the associated energies, the constraint treatment, and the numerical integration strategy, with clear links to the structure of the implementation where this clarifies the mathematics. Numerical results are not the focus here; rather, the intention is to provide a transparent formulation that can be followed and audited line by line.

Motion is assumed planar. The flexible beam undergoes large overall translations and rotations but small strains in its element-local frames. Shear deformation and rotary inertia are retained following Timoshenko kinematics. Gravity acts uniformly along the negative y-direction. Material and section parameters are taken as constant along the beam. A tendon connecting the inner and outer attachment points provides static pretension and viscous action along the cable direction.

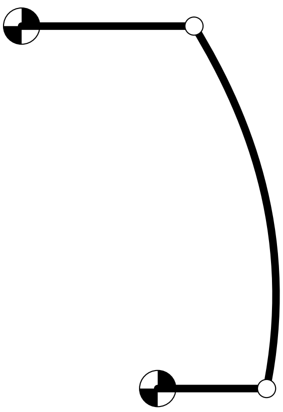


Figure 5.1: Sketch of the desired PACOMA leg after pretension.

All geometric and material parameters used in the planar leg model are listed in Table 5.2. They are consistent with the PACOMA prototype and with the MSC Adams model developed in the project [1, 2].
The goal is to arrive at the modeling of a structure like the one in Figure 5.1, in which the upper rigid bar representing the end effector of PACOMA is horizontal,

Table 5.1: Symbols used in the full flexible multi body simulation model chapter.

Symbol	**Description**	**Units (SI)**
x, y	Coordinates of the fixed inertial frame	m
O, C	Ground pivots of lower and upper rigid links	m
A, B	Connection points between rigid links and flexible beam (beam ends)	m
P_1	Ground hinge of the lower rigid link, $P_1 = (0.150{,}0)^\mathsf{T}$	m
L_0, L_1, L_2, L_3	Vertical offset of upper link base, lengths of lower rigid link, flexible beam, and upper rigid link	m
h, b	Height and width of the rectangular cross-section of the flexible beam	m
N_e, N_n	Number of Timoshenko elements and number of nodes of the flexible beam ($N_n = N_e + 1$)	–
Δx	Nominal element length of the flexible beam, $\Delta x = L_2/N_e$	m
q_1, q_2, q_3	Generalized angles of lower link, baseline beam orientation, and upper link (counterclockwise)	rad

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\dot{q}_i, \ddot{q}_i$	Generalized velocities and accelerations associated with q_i	rad/s, rad/s^2
$\boldsymbol{q}$	Global generalized coordinate vector $[q_1, q_2, q_3, q_d^\mathsf{T}]^\mathsf{T}$	(rad, m)
n	Dimension of the generalized coordinate vector, $n = 3 + n_\mathrm{d}$	–
$\boldsymbol{q}_\mathrm{d}$	Vector of active flexible DOFs of the beam (nodal u, v, θ)	(m, rad)
n_d	Number of active elastic DOFs of the beam	–
`ActiveDOF`	Logical $(N_n \times 3)$ matrix indicating which of (u, v, θ) is active at each node	–
`nf`	Integer $(N_n \times 3)$ map from active nodal DOFs to indices in $\boldsymbol{q}_\mathrm{d}$	–
s_i	Arc abscissa of node i along the beam, $s_i = (i-1)\Delta x$	m
$\bar{\mathbf{x}}_i$	Baseline (undeformed in local sense) position of node i, Eq. (5.4)	m
$\mathbf{x}_i$	Current global position of node i, Eq. (5.5)	m
ψ	Baseline beam direction, $\psi = q_1 + q_2$	rad
ψ_e	Current element direction (co-rotational frame) computed from the chord of element e	rad
$\mathbf{t}_e, \mathbf{n}_e$	Unit tangent and normal vectors of element e in global coordinates	–
u_i, v_i, θ_i	Axial, transverse displacement and cross-section rotation at beam node i (local co-rotational frame)	m, m, rad
$u(x), v(x), \theta(x)$	Continuous axial, transverse, and rotational fields over an element	m, m, rad
ξ	Local natural coordinate of a Gauss point in an element, $\xi \in [-1,1]$	–
$N_1(\xi), N_2(\xi)$	Linear shape functions used to interpolate u, v, θ over a two-node element	–

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\mathbf{R}_e$	3×3 co-rotational mapping from local (u, v, θ) to global (x, y, θ) for one node	–
$\mathbf{T}_e$	6×6 block-diagonal mapping blkdiag$(\mathbf{R}_e, \mathbf{R}_e)$ for a two-node element	–
$\dot{\mathbf{x}}(\xi)$	Global translational velocity at a Gauss point, Eq. (5.8)	m/s
$\dot{\mathbf{x}}_{\text{base}}$	Velocity induced by the motion of the lower-link tip, Eq. (5.9)	m/s
$\dot{\mathbf{x}}_{\text{rigid}}(\xi)$	Velocity induced by the spin of the straight baseline at distance $s(\xi)$, Eq. (5.10)	m/s
$\dot{\mathbf{x}}_{\text{flex}}(\xi)$	Elastic contribution to velocity in the co-rotational frame, mapped to global, Eq. (5.11)	m/s
$\omega(\xi)$	Cross-section angular velocity at a Gauss point, Eq. (5.14)	rad/s
$\varepsilon, \gamma, \kappa$	Axial strain, shear strain, and curvature in the Timoshenko co-rotational formulation, Eq. (5.15)	–, –, 1/m
$\bar{\theta}$	Element-averaged section rotation, Eq. (5.16)	rad
L_e	Element length (current or nominal, used in strain measures)	m
E	Young modulus of the beam material	Pa
G	Shear modulus of the beam material	Pa
A	Cross-sectional area of the flexible beam	m^2
I	Second moment of area of the flexible beam	m^4
κ (Timoshenko)	Shear correction factor in the Timoshenko constitutive law	–
w	Elastic energy density per unit length, Eq. (5.18)	J/m
U_e	Elastic strain energy of one element, Eq. (5.19)	J
m_1, m_3	Masses of the lower and upper rigid links	kg
ρ	Material density of the flexible beam	kg/m^3
g	Gravitational acceleration (acting along negative y)	m/s^2

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\mathcal{K}_{\text{rigid}}$	Kinetic energy of the two rigid links, Eq. (5.20)	J
$\mathcal{K}_{\text{beam}}$	Translational + rotary kinetic energy of the flexible beam, Eq. (5.21)	J
$\mathcal{K}$	Total kinetic energy of rigid links and beam, $\mathcal{K} = \mathcal{K}_{\text{rigid}} + \mathcal{K}_{\text{beam}}$	J
$\mathcal{U}_{g,\text{rigid}}$	Gravitational potential energy of rigid links, Eq. (5.22)	J
$\mathcal{U}_{g,\text{beam}}$	Gravitational potential energy of the flexible beam, Eq. (5.23)	J
$\mathcal{U}_{\text{strain}}$	Total elastic strain energy of the beam, Eq. (5.24)	J
$\mathcal{U}$	Total potential energy, $\mathcal{U} = \mathcal{U}_{g,\text{rigid}} + \mathcal{U}_{g,\text{beam}} + \mathcal{U}_{\text{strain}}$	J
$y_{\text{cm},1}, y_{\text{cm},3}$	Vertical coordinates of the center of mass of links 1 and 3	m
$y(\xi)$	Current vertical coordinate of a Gauss point on the beam	m
J	Jacobian of the mapping from ξ to physical coordinate ($J = \Delta x/2$) in Gauss integration	m
w_g	Gauss quadrature weights (two-point rule, $w_g = 1$)	–
$\mathcal{L}$	Lagrangian of the rigid–flexible system, $\mathcal{L} = \mathcal{K} - \mathcal{U}$	J
$\boldsymbol{E}(q, \dot{q}, \ddot{q})$	Euler–Lagrange operator assembled over all generalized coordinates, Eq. (5.28)	N, N · m
$\mathbf{M}(q)$	Generalized mass matrix, Hessian of kinetic energy w.r.t. $\dot{q}$	kg
$\mathbf{H}(q, \dot{q})$	Non-inertial term in the equations of motion, Eq. (5.29)	N, N · m
$\mathbf{C}(q, \dot{q})$	Velocity-dependent (convective/Coriolis/gyroscopic) part of $\mathbf{H}$	N, N · m
$\mathbf{g}(q)$	Gravity contribution in generalized coordinates	N, N · m
$\mathbf{k}_{\text{int}}(q)$	Internal elastic forces from the strain energy	N, N · m
$\mathbf{K}_t$	Tangent stiffness matrix, $\partial\mathbf{H}/\partial q$	N/m

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\mathbf{C}_t$	Tangent damping matrix, $\partial \mathbf{H}/\partial \dot{q}$	$\mathrm{N} \cdot \mathrm{s/m}$
$\mathbf{k}_e^{\mathrm{loc}}, \mathbf{m}_e^{\mathrm{loc}}$	Local element stiffness and mass matrices in co-rotational frame	N/m, kg
$\mathbf{k}_{\mathrm{ax}}$	Axial part of local stiffness (unit EA/L_e factor), Eq. (5.32)	–
$\mathbf{k}_{\mathrm{fl}}$	Bending–shear part of local stiffness (unit EI/L_e^3 factor)	–
$\mathbf{m}_{\mathrm{tr}}$	Consistent translational mass block (unit $\rho A L_e$ factor)	–
$\mathbf{m}_{\mathrm{rot}}$	Rotary inertia block (unit $\rho I L_e$ factor)	–
ϕ (shear index)	Shear flexibility parameter $\phi = 12EI/(\kappa G A L_e^2)$	–
c_e, s_e	Cosine and sine of the element direction ψ_e	–
$\mathbf{k}_e, \mathbf{m}_e$	Global element stiffness and mass matrices, Eq. (5.35)	N/m, kg
$\mathbf{A}_e$	Boolean scatter matrix mapping element DOFs into global DOFs	–
$\mathbf{K}, \mathbf{M}$	Assembled global stiffness and mass matrices of the beam (in the flexible DOF space)	N/m, kg
$\boldsymbol{p}_B(q)$	Current global position of the flexible beam tip	m
$\boldsymbol{p}_3(q_3)$	Current global position of the upper-link tip, Eq. (5.38)	m
$\boldsymbol{\phi}(q)$	Loop-closure constraint, mismatch $\boldsymbol{p}_B(q) - \boldsymbol{p}_3(q_3)$	m
$\mathbf{J}_c(q)$	Constraint Jacobian, $\partial \boldsymbol{\phi}/\partial q$	–
$\boldsymbol{\lambda}$	Lagrange multipliers associated with loop-closure; reaction forces at the tip	N
ζ, ω_B	Baumgarte stabilization damping ratio and angular frequency in the acceleration-level constraint, Eq. (5.42)	–, s^{-1}
z_B, w_B	Code-level notation for ζ, ω_B (Baumgarte parameters)	–, s^{-1}

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
$\mathbf{B}$	Lower-left block in the KKT system, linearization of stabilized constraints	–
$\mathbf{S}_t$	Effective tangent matrix in generalized-α iterations, Eq. (5.52)	N/m
$\mathbf{r}_m, \mathbf{r}_c$	Residuals of the momentum equation and constraint equation in the KKT system	N, m
$\mathbf{S}_\lambda$	Schur-complement matrix for multipliers in the KKT system, Eq. (5.55)	–
$\boldsymbol{p}_A(q)$	Current global position of the tendon attachment on link 1	m
$\boldsymbol{r}(q)$	Relative vector from A to B, $\boldsymbol{r} = \boldsymbol{p}_B - \boldsymbol{p}_A$	m
$L(q)$	Current tendon length, $L(q) = \lVert\boldsymbol{r}(q)\rVert$	m
L_0	Tendon rest length used in the spring law	m
$\boldsymbol{e}(q)$ or $\hat{\boldsymbol{e}}$	Unit direction of the tendon, $\boldsymbol{e} = \boldsymbol{r}/\lVert\boldsymbol{r}\rVert$	–
$\boldsymbol{J}_A, \boldsymbol{J}_B$	Jacobians of endpoints A and B with respect to generalized coordinates	–
$\boldsymbol{J}_{\mathrm{rel}}$	Relative Jacobian $\boldsymbol{J}_B - \boldsymbol{J}_A$, mapping $\dot{q}$ to endpoint relative velocity	–
$F(q, \dot{q})$	Scalar tendon force from spring–damper law, Eq. (5.46)	N
k (tendon)	Tendon stiffness in the spring–damper model	N/m
c (tendon)	Tendon viscous coefficient in the spring–damper model	N · s/m
$\boldsymbol{f}$	Global tendon force vector at endpoints ($\boldsymbol{f} = F\,\boldsymbol{e}$)	N
$\boldsymbol{Q}_{\mathrm{ten}}$	Generalized force vector contributed by the tendon, Eq. (5.47)	N, N · m
$\mathbf{K}_{\mathrm{ten}}$	Elastic tangent of the tendon force, Eq. (5.48)	N/m
τ	Applied generalized load vector (actuation, external torques/forces)	N, N · m

Continued on next page

Table 5.1: Symbols used in the full flexible multi body simulation model chapter. (Continued)

Symbol	**Description**	**Units (SI)**
ρ_∞	Target spectral radius at infinity for generalized-α; controls high-frequency numerical damping	–
α_m, α_f	Generalized-α parameters weighting accelerations and forces, Eq. (5.50)	–
γ, β	Newmark-like parameters derived from α_m, α_f, Eq. (5.50)	–
Δt	Time step of the integration scheme	s
q_{eff}^{n+1}	Algorithmic configuration state used to evaluate residuals, Eq. (5.51)	(rad, m)
$\dot{q}_{\text{eff}}^{n+1}, \ddot{q}_{\text{eff}}^{n+1}$	Algorithmic velocity and acceleration states in generalized-α	(rad/s, m/s), (rad/s^2, m/s^2)
β_p, γ_p	Effective generalized-α coefficients multiplying $\mathbf{M}$ and $\mathbf{C}_{\text{tot}}$ in $\mathbf{S}_t$, Eq. (5.52)	–
$\mathbf{C}_{\text{Ray}}$	Rayleigh damping matrix, $\mathbf{C}_{\text{Ray}} = \alpha_R \mathbf{M}_{ff} + \beta_R \mathbf{K}_{ff}$	N · s/m
α_R, β_R	Rayleigh mass- and stiffness-proportional damping coefficients, Eq. (5.58)	s^{-1}, s
$\mathbf{C}_{\text{tot}}$	Total damping matrix in the iteration, $\mathbf{C}_{\text{tot}} = \mathbf{C}_t + \mathbf{C}_{\text{Ray}}$	N · s/m
$\mathbf{M}_{ff}, \mathbf{K}_{ff}$	Mass and stiffness submatrices restricted to flexible DOFs	kg, N/m
ζ	Target viscous damping ratio for the first bending modes in the Rayleigh model	–
ω_1, ω_2	Circular frequencies of the first two bending modes of the flexible beam	rad/s
$\zeta(\omega)$	Resulting Rayleigh damping ratio at generic circular frequency ω	–

5.2 Geometry, coordinates, and structure of the unknowns

A fixed inertial frame (x, y) is adopted. The lower rigid link of length L_1 is grounded at the pivot

$$O = \begin{bmatrix} 0.150 \\ 0 \end{bmatrix} \text{ m}$$

and is described by a single generalized coordinate q_1 (positive counterclockwise). The upper rigid link of length L_3 is grounded at

$$C = \begin{bmatrix} 0 \\ L_0 \end{bmatrix}$$

and is described by q_3. The flexible beam of length L_2 connects the tip of link 1 to the tip of link 3. It is discretized into N_e two-node Timoshenko elements of nominal length $\Delta x = L_2/N_e$. It has a rectangular cross-section, with height $h = 1.2$ mm and a width $b = 77$ mm, as stated in the PACOMA report[1].

A baseline orientation parameter q_2 encodes the mean direction of the beam centerline prior to elastic deformation. The beam nodal elastic unknowns are, at each node, the local axial displacement u, the local transverse displacement v, and the cross-section rotation θ. The global vector of generalized coordinates is then ordered as

$$q = \begin{bmatrix} q_1 & q_2 & q_3 & q_d^\top \end{bmatrix}^\top \in \mathbb{R}^n, \qquad n = 3 + n_d,$$

where $q_d \in \mathbb{R}^{n_d}$ collects the active degrees of freedom of the flexible beam's nodes. This ordering separates the minimal rigid coordinates from the deformable ones and facilitates assembly and constraint handling.

Figure 5.2 shows the initial geometry of the rigid-flexible model. L_1 is the length of the lower rigid bar connected to the actuator. L_3 is the length of the upper rigid bar, representing the end effector. L_2 is the length of the titanium leaf spring before buckling due to the pretension applied by the tendon, as indicated in Table 1.1.

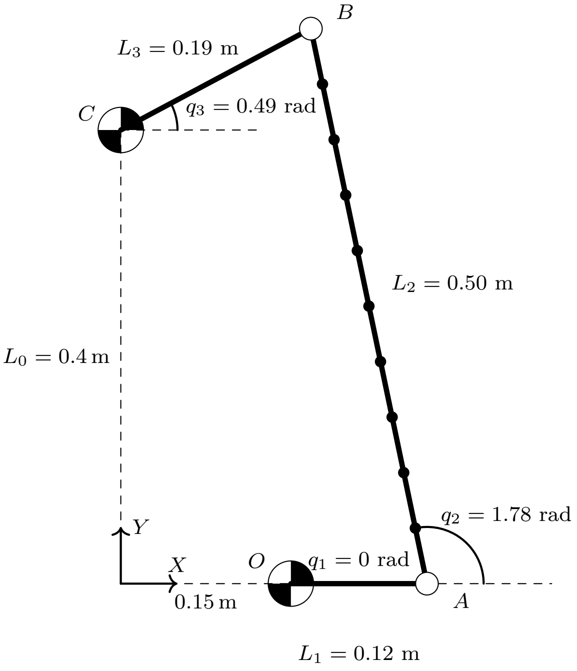


Figure 5.2: Rigid-flexible model's initial geometry.

Table 5.2: Fixed geometric, material, and numerical parameters used in the full flexible multi body simulation model.

Symbol	Description	Value	Units (SI)
O	Ground pivot of the lower rigid link in the planar leg model	(0.000, 0.000)	m
C	Ground pivot of the upper rigid link	–	m
P_1	Ground hinge of the lower rigid link (used in the planar reduction)	(0.150, 0.000)	m
L_0	Vertical offset between the two ground hinges	0.40	m
L_1	Length of the lower rigid link	0.12	m
L_2	Length of the flexible beam (coupler)	0.50	m
L_3	Length of the upper rigid link	0.19	m
b	Width of the rectangular cross-section of the flexible beam	$77 \cdot 10^{-3}$	m

Continued on next page

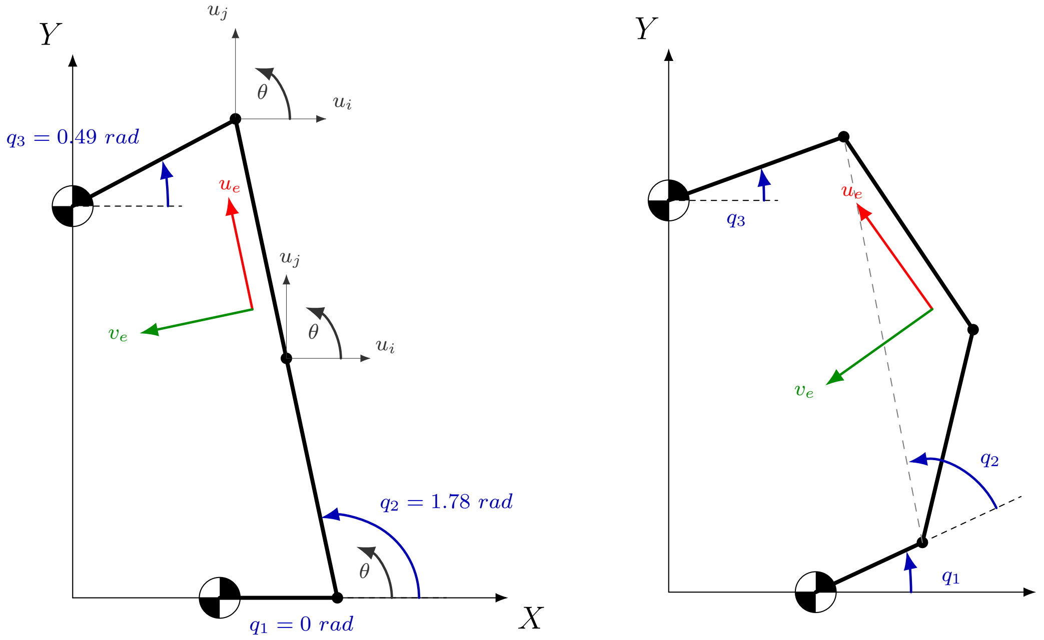
Table 5.2: Fixed geometric, material, and numerical parameters used in the full flexible multi body simulation model. (Continued)

Symbol	**Description**	**Value**	**Units (SI)**
h	Height (thickness) of the rectangular cross-section of the flexible beam	$1.2 \cdot 10^{-3}$	m
N_e	Number of Timoshenko elements discretizing the flexible beam	10 (or 4)	–
N_n	Number of nodes of the flexible beam ($N_n = N_e + 1$)	11 (or 5)	–
E	Young's modulus of the beam material	$120 \cdot 10^9$	Pa
ν	Poisson's ratio of the beam material	0.35	–
G	Shear modulus of the beam material	$4.444 \cdot 10^{10}$	Pa
k (shear)	Shear correction factor in the Timoshenko constitutive law	5/6	–
ρ	Material density of the flexible beam	4502	kg/m^3
m_1	Mass of the lower rigid link	1	kg
m_3	Mass of the upper rigid link	1	kg
g	Gravitational acceleration (negative y–direction)	9.81	m/s^2
Δt	Time step size in the generalized–α integration of the leg model	$1 \cdot 10^{-3}$	s
ρ_∞	Target spectral radius at infinity (generalized–α scheme)	0.8	–

5.3 Modeling assumptions and rationale

A co-rotational beam formulation is adopted, following [20, 16]. The central idea is to describe each finite element in a local frame that follows the current element chord, so that large rigid motions are filtered out, while the elastic measures u, v, θ remain relatively small. This approach preserves the familiar local Timoshenko matrices [13, 14], confines geometric nonlinearities to a limited number of terms, and makes explicit where the dependence on the baseline direction $q_1 + q_2$ enters and where the elastic corrections act. The shear correction factor κ is taken according to standard corrections for prismatic sections [5, 12].

Figure 5.3 shows the initial configuration, in which the flexible beam is straight and an arbitrary configuration in which a pretension applied between the two end



(a) Initial configuration

(b) Arbitrary configuration

Figure 5.3: Rigid-flexible model's active degrees of freedom. Flexible beam discretized with 2 elements just as example.

nodes tends to buckle the flexible beam. The three main angles q_1, q_2, q_3 are shown in blue, and the degrees of freedom active for each node are shown in gray. The first node (connection between the lower rigid bar and the flexible beam) has only the degree of freedom of rotation active, since the longitudinal and transverse deformations are constrained by the rotational joint present at this point.
The holonomic loop-closure is expressed by the vector-valued map

$$\boldsymbol{\phi}(q) \;=\; \boldsymbol{p}_B(q) \;-\; \boldsymbol{p}_3(q_3) \;\in\; \mathbb{R}^2 \tag{5.1}$$

which measures in global coordinates the translational mismatch between the flexible beam tip B and the tip of the upper rigid link. The constraint $\boldsymbol{\phi}(q) = \mathbf{0}$ enforces coincidence of the two points [21, 17].
The map $\boldsymbol{p}_B(q)$ returns the current global position of the beam tip as a function of all generalized coordinates q. Let $P_1 = (0.150,\ 0)^\top$, L_1 the lower-link length, L_2 the beam length, and $\psi = q_1 + q_2$ the baseline beam direction. Denote by $\boldsymbol{t}_{N_e}(q)$ and $\boldsymbol{n}_{N_e}(q)$ the unit tangent and normal of the last (co-rotational) beam element, evaluated from the current chord of that element, and by u_{N+1}, v_{N+1} the axial and transverse nodal DOFs at the beam tip (active in this model). Then a convenient

explicit representation is

$$\boldsymbol{p}_B(q) \;=\; \underbrace{P_1 + L_1\begin{bmatrix}\cos q_1\\ \sin q_1\end{bmatrix}}_{\text{position of the lower-link tip}} \;+\; \underbrace{L_2\begin{bmatrix}\cos\psi\\ \sin\psi\end{bmatrix}}_{\text{baseline beam reach}} \;+\; \underbrace{u_{N+1}\,\boldsymbol{t}_{N_e}(q) + v_{N+1}\,\boldsymbol{n}_{N_e}(q)}_{\text{elastic correction at the tip}}.$$

In the implementation this quantity is obtained by the routine `current_geom_from_q`, which builds the full nodal geometry using the same co-rotational mapping; the expression above makes explicit the dependence on q_1, q_2 and on the tip DOFs u_{N+1}, v_{N+1} [21].
The map $\boldsymbol{p}_3(q_3)$ returns the current global position of the tip of the upper rigid link of length L_3, whose base is fixed at $C = [0,\, L_0]^\top$. It depends only on the single generalized coordinate q_3 and reads

$$\boldsymbol{p}_3(q_3) \;=\; \begin{bmatrix} L_3\cos q_3\\ L_0 + L_3\sin q_3\end{bmatrix}. \tag{5.2}$$

The Jacobian of the constraint, $\boldsymbol{J}_c(q) = \partial\boldsymbol{\phi}/\partial q \in \mathbb{R}^{2\times n}$, follows as

$$\boldsymbol{J}_c(q) \;=\; \frac{\partial \boldsymbol{p}_B}{\partial q}(q) \;-\; \underbrace{\frac{\partial \boldsymbol{p}_3}{\partial q}(q)}_{=\left[\begin{smallmatrix}\mathbf{0} & \mathbf{0} & \begin{smallmatrix}-L_3\sin q_3\\ L_3\cos q_3\end{smallmatrix} & \mathbf{0}\end{smallmatrix}\right]} \tag{5.3}$$

with the nonzero columns of $\partial\boldsymbol{p}_B/\partial q$ located at q_1, q_2 and at the indices corresponding to the active tip DOFs u_{N+1}, v_{N+1} (and, through the co-rotational frame, at additional elastic DOFs that affect the last-element orientation). The projection routine `project_loop` enforces $\boldsymbol{\phi}(q) = \mathbf{0}$ (and $\boldsymbol{J}_c\dot{q} = \mathbf{0}$ at the velocity level) [10, 9, 22], thereby determining the absolute position of the beam tip. In practice, this projection drives the pair (u_{N+1}, v_{N+1}) to the values required to satisfy closure, while the section rotation θ_{N+1} remains free.

```matlab
function [q_corr, dq_corr] = project_loop(q, dq, prop, L, fixed_idx, nf)
% Position projection: solve phi(q)=0 by weighted min-norm correction.
% Velocity projection: enforce Jc(q)*dq=0.
%
% Inputs:
% q, dq current state
% prop, L model parameters (for generated loop_phi / loop_Jc)
% fixed_idx logical mask of DOFs to keep fixed (Dirichlet)
% nf numbering matrix for active (u,v,theta); used to clamp u,v steps
%
```

```
% Outputs:
% q_corr, dq_corr projected position and velocity

% Weights: very large for fixed DOFs -> minimal motion there
Wdiag = ones(n,1);
Wdiag(fixed_idx) = 1e12;

% Small safeguards
maxit = 8; tol = 1e-12;

% ---- Position-level projection: phi(q)=0 ----
for it = 1:maxit
    phi = loop_phi(prop, L, q); % 2x1
    if norm(phi,2) < tol,
        break;
    end

    Jc = loop_Jc(prop, L, q);             % 2xn
    % WinvJt = W^{-1} * Jc'  (implemented with element-wise division by Wdiag)
    WinvJt = Jc.' ./ Wdiag;                % n x 2
    S = Jc * WinvJt;                  % 2 x 2

    % Robust 2x2 inverse with light regularization if needed
    detS = S(1,1)*S(2,2) - S(1,2)*S(2,1);
    if ~isfinite(detS) || abs(detS) < 1e-12
        S = S + 1e-9*eye(2);
        detS = S(1,1)*S(2,2) - S(1,2)*S(2,1);
    end
    Sinv = [ S(2,2), -S(1,2); -S(2,1), S(1,1) ] / detS;

    dq_pos = - WinvJt * (Sinv * phi);     % n x 1

    % Optional clamp on elastic u,v increments (stability aid)
    if ~isempty(nf)
        rigid_off = 3;
        uv_step_max = 1e-3;               % m per iteration
        iu = nf(:,1); iu = iu(iu>0); idx_u = rigid_off + iu;
        iv = nf(:,2); iv = iv(iv>0); idx_v = rigid_off + iv;
        dq_pos(idx_u) = max(min(dq_pos(idx_u), +uv_step_max), -uv_step_max);
        dq_pos(idx_v) = max(min(dq_pos(idx_v), +uv_step_max), -uv_step_max);
    end

    dq_pos(fixed_idx) = 0;                % honor fixed DOFs
    q = q + dq_pos;
    if norm(dq_pos,2) < tol,
        break;
```

```
    end
end

% ---- Velocity-level projection: Jc(q)dq = 0 ----
Jc = loop_Jc(prop, L, q); % 2xn
G = JcJc.'; % 2x2
detG = G(1,1)*G(2,2) - G(1,2)G(2,1);
if ~isfinite(detG) || abs(detG) < 1e-12
    G = G + 1e-9eye(2);
    detG = G(1,1)*G(2,2) - G(1,2)*G(2,1);
end
Ginv = [ G(2,2), -G(1,2); -G(2,1), G(1,1) ] / detG;

dq = dq - Jc.' * ( Ginv * (Jc*dq) );
dq(fixed_idx) = 0;

q_corr = q;
dq_corr = dq;
end
```

Listing 5.1: Constraint projection at position and velocity levels (`project_loop`).

```
function geom = current_geom_from_q(L1, q, dx, Ne, nf)
% Build current global coordinates of the (Ne+1) beam nodes.
% Baseline: tip of link 1 + straight reach at angle psi = q1+q2.
% Elastic correction: local (u,v) mapped directly in global frame at nodes.

P1 = [0.150; 0.0];
q1 = q(1);
q2 = q(2);
psi = q1 + q2;

base = P1 + [L1*cos(q1); L1*sin(q1)];

geom = zeros(Ne+1, 2);

for i = 1:(Ne+1)
    s = (i-1)dx;
    x0 = base + s*[cos(psi); sin(psi)]; % baseline node position

    % Local nodal DOFs (active according to nf)
    ui = 0; vi = 0;
    iu = nf(i,1);  if iu>0, ui = q(3+iu); end
    iv = nf(i,2);  if iv>0, vi = q(3+iv); end

    xi = x0 + [ui; vi];      % current global node position
    geom(i,:) = xi.';
end
```

```
end
```

Listing 5.2: Co-rotational reconstruction of current beam-node positions (`current_geom_from_q`).

5.4 Mesh, active degrees of freedom, and numbering

Let `ActiveDOF` $\in \{0,1\}^{(N_e+1)\times 3}$ be the logical matrix that indicates which of the triplet (u, v, θ) is active at each beam node. Its companion `nf` $\in \mathbb{N}_0^{(N_e+1)\times 3}$ maps each active entry to a unique global index among the elastic coordinates. A simple numbering routine increments a counter only when `ActiveDOF`$(i, k) = 1$. This mapping, although basic, is the backbone of a clean assembly because it prevents the introduction of spurious zero rows or columns and allows dispersion from element to global without special cases at the ends.

5.5 Co-rotational kinematics of the beam

Let the arc abscissa of node i be $s_i = (i-1)\,\Delta x$. The baseline (undeformed-in-the-local-sense) position is

$$\bar{\mathbf{x}}_i(q_1, q_2) \;=\; \begin{bmatrix} 0.150 \\ 0 \end{bmatrix} \;+\; L_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix} \;+\; s_i \begin{bmatrix} \cos(q_1+q_2) \\ \sin(q_1+q_2) \end{bmatrix}. \tag{5.4}$$

The current position augments the baseline by the nodal deformations expressed in the local co-rotational frame of the element e that contains node i:

$$\mathbf{x}_i \;=\; \bar{\mathbf{x}}_i \;+\; u_i\,\mathbf{t}_e \;+\; v_i\,\mathbf{n}_e, \qquad \mathbf{t}_e = \begin{bmatrix} \cos\psi_e \\ \sin\psi_e \end{bmatrix}, \quad \mathbf{n}_e = \begin{bmatrix} -\sin\psi_e \\ \cos\psi_e \end{bmatrix}, \tag{5.5}$$

where ψ_e is the current element direction obtained from the chord joining the two current end nodes of element e as shown in Figure 5.3. A small regularization $O(10^{-9}\Delta x)$ can be added under the square root in the chord length to avoid a division by zero, as a protection from some unwanted problem.

For a two-node Timoshenko element the local shape matrix reads

$$\mathbf{N}(\xi) \;=\; \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 \end{bmatrix} \tag{5.6}$$

$$N_1(\xi) = \frac{1-\xi}{2}, \qquad N_2(\xi) = \frac{1+\xi}{2}, \qquad \xi \in [-1,1].$$

Let $c = \cos\psi_e$ and $s = \sin\psi_e$. The 3×3 co-rotational mapping that converts local nodal triplets (u, v, θ) into global increments (x, y, θ) for a single node is

$$\mathbf{R}_e = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

For the two-node element the 6×6 block-diagonal map is

$$\mathbf{T}_e = \operatorname{blkdiag}(\mathbf{R}_e, \mathbf{R}_e) = \begin{bmatrix} c & -s & 0 & 0 & 0 & 0 \\ s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -s & 0 \\ 0 & 0 & 0 & s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.7}$$

This matrix sends the local element vector $\mathbf{q}_e^{\text{loc}} = [u_j, v_j, \theta_j, u_k, v_k, \theta_k]^\top$ to its global counterpart via $\mathbf{q}_e^{\text{glob}} = \mathbf{T}_e\, \mathbf{q}_e^{\text{loc}}$.

Consider a Gauss point inside element e at abscissa

$$s(\xi) = s_j + \frac{1+\xi}{2}\,\Delta x,$$

where s_j is the arc position of node j and Δx is the nominal element length. Its global translational velocity is the sum of three physically distinct contributions:

$$\begin{aligned} \dot{\mathbf{x}}(\xi) = {} & \underbrace{\dot{\mathbf{x}}_{\text{base}}}_{\text{motion of the lower-link tip}} \\ & + \underbrace{\dot{\mathbf{x}}_{\text{rigid}}(\xi)}_{\text{spin of the straight baseline at distance } s(\xi)} \\ & + \underbrace{\dot{\mathbf{x}}_{\text{flex}}(\xi)}_{\text{elastic rate in the local frame, mapped to global}} . \end{aligned} \tag{5.8}$$

Each term can be written in closed form:

$$\dot{\mathbf{x}}_{\text{base}} = \frac{d}{dt}\left(\begin{bmatrix} 0.150 \\ 0 \end{bmatrix} + L_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix}\right) = L_1 \dot{q}_1 \begin{bmatrix} -\sin q_1 \\ \cos q_1 \end{bmatrix} \tag{5.9}$$

$$\dot{\mathbf{x}}_{\text{rigid}}(\xi) = \frac{d}{dt}\left(s(\xi) \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix}\right) = s(\xi)\,(\dot{q}_1 + \dot{q}_2) \begin{bmatrix} -\sin\psi \\ \cos\psi \end{bmatrix}, \text{ with } \psi = q_1 + q_2 \tag{5.10}$$

$$\dot{\mathbf{x}}_{\text{flex}}(\xi) = \mathbf{R}_2(\psi_e) \begin{bmatrix} \dot{u}(\xi) \\ \dot{v}(\xi) \end{bmatrix}, \text{ with } \mathbf{R}_2(\psi_e) = \begin{bmatrix} \cos\psi_e & -\sin\psi_e \\ \sin\psi_e & \cos\psi_e \end{bmatrix} \tag{5.11}$$

with the local elastic rates at the Gauss point obtained from the nodal rates as

$$\dot{u}(\xi) = \begin{bmatrix} N_1(\xi) & 0 & 0 & N_2(\xi) & 0 & 0 \end{bmatrix} \dot{\mathbf{q}}_e^{\mathrm{loc}}, \tag{5.12}$$

$$\dot{v}(\xi) = \begin{bmatrix} 0 & N_1(\xi) & 0 & 0 & N_2(\xi) & 0 \end{bmatrix} \dot{\mathbf{q}}_e^{\mathrm{loc}}. \tag{5.13}$$

Expression (5.8) separates kinematically the transport of the Gauss point by the rigid motion of the lower link (5.9), the spin of the straight baseline induced by $\dot{\psi} = \dot{q}_1 + \dot{q}_2$ at distance $s(\xi)$ (5.10), and the elastic rate in the element co-rotational frame mapped to global coordinates (5.11). In the small-strain co-rotational setting the time variation of the local frame itself, which would generate higher-order convective terms proportional to $\dot{\psi}_e\, u(\xi)$ and $\dot{\psi}_e\, v(\xi)$, is neglected in the translational velocity because u and v are small by construction.

The angular velocity entering the rotary kinetic energy follows from the superposition of the baseline spin and the local interpolated section-rotation rate,

$$\omega(\xi) \;=\; (\dot{q}_1 + \dot{q}_2) \;+\; \underbrace{\begin{bmatrix} 0 & 0 & N_1(\xi) & 0 & 0 & N_2(\xi) \end{bmatrix}}_{=\,\mathbf{N}^{(3)}(\xi)} \dot{\mathbf{q}}_e^{\mathrm{loc}}. \tag{5.14}$$

5.6 Strain measures and constitutive density

In the co-rotational Timoshenko setting with a von Kármán enrichment [23, 16, 15], axial strain is augmented to capture moderate rotations while keeping small strains. Let $x \in [0, L_e]$ denote the element arc coordinate in the local (co-rotational) frame, and let $u(x)$, $v(x)$, $\theta(x)$ be, respectively, the axial displacement, the transverse displacement, and the cross-section rotation in that frame. The adopted measures are

$$\varepsilon(x) \;=\; \frac{du}{dx}(x) \;+\; \frac{1}{2}\left(\frac{dv}{dx}(x)\right)^2 \quad ; \quad \gamma(x) \;=\; \frac{dv}{dx}(x) \;-\; \bar{\theta} \quad ; \quad \kappa(x) \;=\; \frac{d\theta}{dx}(x) \tag{5.15}$$

where $\bar{\theta}$ is the element-averaged section rotation. In this work the average is taken as the nodal mean

$$\bar{\theta} \;=\; \tfrac{1}{2}\Big(\theta_j + \theta_k\Big) \tag{5.16}$$

which is constant over the element and coincides with the definition used in the implementation.

With the two-node interpolation adopted for the triplet (u, v, θ),

$$u(x) = N_1(x)\, u_j + N_2(x)\, u_k$$

$$v(x) = N_1(x)\, v_j + N_2(x)\, v_k$$

$$\theta(x) = N_1(x)\,\theta_j + N_2(x)\,\theta_k$$

the element-wise derivatives are constant:

$$\frac{du}{dx} \;=\; \frac{u_k - u_j}{L_e}, \qquad \frac{dv}{dx} \;=\; \frac{v_k - v_j}{L_e}, \qquad \frac{d\theta}{dx} \;=\; \frac{\theta_k - \theta_j}{L_e}. \tag{5.17}$$

Substituting (5.17) and (5.16) into (5.15) yields constant ε_x, γ, and κ over the element (note that the nonlinearity enters through the quadratic term $\frac{1}{2}(v_x)^2$, but v_x itself is constant). This property implies that a two-point Gauss rule integrates exactly the element energy, as used in the codes [13, 14].
The Timoshenko constitutive density in the co-rotational frame reads

$$w \;=\; \tfrac{1}{2}\,EA\,\varepsilon^2 \;+\; \tfrac{1}{2}\,\kappa GA\,\gamma^2 \;+\; \tfrac{1}{2}\,EI\,\kappa^2 \tag{5.18}$$

with E the Young modulus, G the shear modulus, A the cross-sectional area, I the second moment of area, and k the shear-correction factor for the chosen prismatic section [5, 14, 13]. Since the integrand is constant under (5.17), the element elastic energy takes the compact form

$$U_e \;=\; \int_0^{L_e} w\,dx \;=\; L_e\left[\tfrac{1}{2}\,EA\,\varepsilon_x^2 \;+\; \tfrac{1}{2}\,\kappa GA\,\gamma^2 \;+\; \tfrac{1}{2}\,EI\,\kappa^2\right]. \tag{5.19}$$

This expression is algebraically equivalent to the one assembled in the implementation, where the same measures ε, γ, κ are formed from nodal differences divided by L_e and the energy is accumulated as $U_e\,\Delta x$.

5.6.1 Modeling remarks.

(i) The von Kármán axial strain $\varepsilon = \frac{du}{dx} + \frac{1}{2}\left(\frac{dv}{dx}\right)^2$ supplies the leading-order geometric stiffening that is appropriate for operating configurations with moderate rotations but small strains in the co-rotational frame. (ii) The shear measure $\gamma = \frac{dv}{dx} - \bar{\theta}$ is the standard Timoshenko shear, which allows cross-sections to remain rigid (no warping) without being constrained to remain orthogonal to the deformed centroidal axis. (iii) The factor k accounts for the nonuniform shear distribution over the section; for rectangular sections the canonical $k = 5/6$ is a widely accepted approximation.

5.7 Energies and their physical content

Each rigid link is modeled as a slender, uniform rod rotating about a revolute joint at one end. The kinetic energy of a rod of length L and mass m rotating with angular speed $\dot{q}$ about its end is

$$\mathcal{K} = \tfrac{1}{2} I_{\text{end}} \dot{q}^2, \qquad I_{\text{end}} = \tfrac{1}{3} m L^2,$$

hence $\mathcal{K} = \frac{1}{6} m L^2 \dot{q}^2$. Applying this to links 1 and 3 gives

$$\mathcal{K}_{\text{rigid}} \;=\; \tfrac{1}{6}\, m_1 L_1^2\, \dot{q}_1^2 \;+\; \tfrac{1}{6}\, m_3 L_3^2\, \dot{q}_3^2. \tag{5.20}$$

The flexible beam contributes translational and rotary inertia. At a Gauss point inside element e the translational velocity $\dot{\mathbf{x}}(\xi)$ is the global time derivative of the point position, which—within the co-rotational setting—is naturally decomposed as in Eq. (5.8). The cross-section angular velocity $\omega(\xi)$ follows from Eq. (5.14). Using density ρ, area A, and area moment of inertia I, the kinetic energy density at that point is

$$\tfrac{1}{2}\rho A \|\dot{\mathbf{x}}(\xi)\|^2 \;+\; \tfrac{1}{2}\rho I\, \omega(\xi)^2.$$

Integrating over the element with the reference abscissa $\xi \in [-1{,}1]$ and the affine map $x(\xi)$ yields the Jacobian $J = \Delta x/2$. Using two-point Gauss quadrature (weights $w_g = 1$) and summing over all elements gives

$$\mathcal{K}_{\text{beam}} \;=\; \sum_{e=1}^{N_e} \sum_{g=1}^{2} \Big[\tfrac{1}{2}\rho A\, \|\dot{\mathbf{x}}\|^2 \;+\; \tfrac{1}{2}\rho I\, \omega^2\Big]_{(e,g)} J\, w_g \tag{5.21}$$

$$J = \Delta x/2 \;\;;\;\; w_g = 1.$$

The use of the two-point rule is consistent with the linear interpolation adopted for (u, v, θ) and with the co-rotational mapping (see, e.g., [13, 14, 5]).
The gravitational potential energy of link 1 is $m_1 g\, y_{\text{cm},1}$ with the center-of-mass ordinate $y_{\text{cm},1} = \frac{L_1}{2} \sin q_1$; similarly for link 3, whose base is offset by L_0 in the vertical direction, $y_{\text{cm},3} = L_0 + \frac{L_3}{2} \sin q_3$. Therefore the gravitational potential energy of the rigid links is

$$\mathcal{U}_{g,\text{rigid}} \;=\; m_1 g\, \tfrac{L_1}{2}\, \sin q_1 \;+\; m_3 g \left(L_0 + \tfrac{L_3}{2}\, \sin q_3\right). \tag{5.22}$$

For the flexible beam, the gravitational density is $\rho A g\, y$, where $y(\xi)$ is the current global vertical coordinate of the Gauss point obtained from the co-rotational reconstruction. With the same two-point rule used for kinetics,

$$\mathcal{U}_{g,\text{beam}} \;=\; \sum_{e=1}^{N_e} \sum_{g=1}^{2} \rho A\, g\, y(\xi)\, J\, w_g. \tag{5.23}$$

This expressions mirrors the implementation in which $y(\xi)$ comes from `current_geom_from_q`.
Let w denote the constitutive energy density in the element's co-rotational frame,

$$w \;=\; \tfrac{1}{2} EA\, \varepsilon^2 \;+\; \tfrac{1}{2} kGA\, \gamma^2 \;+\; \tfrac{1}{2} EI\, \kappa^2,$$

with strain measures $(\varepsilon, \gamma, \kappa)$ defined in Eqs. (5.15)–(5.17). The beam strain energy is the sum of elemental contributions

$$\mathcal{U}_{\text{strain}} \;=\; \sum_{e=1}^{N_e} \int_0^{\Delta x} w\, dx. \tag{5.24}$$

Since $\frac{du}{dx}$, $\frac{dv}{dx}$, and $\frac{d\theta}{dx}$ are element-wise constant with the adopted interpolation, a two-point Gauss scheme evaluates $\mathcal{U}_{\text{strain}}$ exactly (see [16, 15, 13]).
The total kinetic and potential energies are

$$\mathcal{K} \;=\; \mathcal{K}_{\text{rigid}} + \mathcal{K}_{\text{beam}}, \qquad \mathcal{U} \;=\; \mathcal{U}_{g,\text{rigid}} + \mathcal{U}_{g,\text{beam}} + \mathcal{U}_{\text{strain}}.$$

The Lagrangian is defined as the difference

$$\mathcal{L}(q, \dot{q}) \;=\; \mathcal{K}(q, \dot{q}) \;-\; \mathcal{U}(q), \tag{5.25}$$

and serves as the starting point for deriving the equations of motion via the Euler–Lagrange equations with holonomic constraints. In compact form one obtains the standard multibody structure:

$$\mathbf{M}(q)\,\ddot{q} + \mathbf{H}(q, \dot{q}) = \tau \tag{5.26}$$

$$\boldsymbol{\phi}(q) = \mathbf{0} \tag{5.27}$$

where $\mathbf{M}$ is the configuration-dependent mass matrix, $\mathbf{H}$ collects Coriolis/centrifugal, gravity, and internal elastic forces, τ denotes applied generalized loads, and $\boldsymbol{\phi}$ encodes the loop closure; see [21, 14].

5.8 Euler–Lagrange operator, mass matrix, and non-inertial term

Let $\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q)$ be the Lagrangian of the rigid–flexible leg. For each generalized coordinate q_i the Euler–Lagrange operator is

$$E_i(q, \dot{q}, \ddot{q}) \;=\; \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) \;-\; \frac{\partial \mathcal{L}}{\partial q_i}. \tag{5.28}$$

Collecting the components into $\boldsymbol{E}(q, \dot{q}, \ddot{q}) \in \mathbb{R}^n$ and introducing the applied generalized loads τ, the unconstrained equations of motion read $\boldsymbol{E}(q, \dot{q}, \ddot{q}) = \tau$.
It is convenient to isolate the part that multiplies accelerations from the remaining effects. The generalized mass matrix and the non-inertial term are defined by

$$\boldsymbol{M}(q) \;=\; \frac{\partial \boldsymbol{E}}{\partial \ddot{q}}(q, \dot{q}, \ddot{q}), \qquad \boldsymbol{H}(q, \dot{q}) \;=\; \boldsymbol{E}(q, \dot{q}, \ddot{q}) \;-\; \boldsymbol{M}(q)\,\ddot{q}, \tag{5.29}$$

so that the standard form

$$\boldsymbol{M}(q)\,\ddot{q} \;+\; \boldsymbol{H}(q,\dot{q}) \;=\; \tau \tag{5.30}$$

is obtained. Since $\mathcal{L} = \mathcal{K} - \mathcal{U}$ and $\mathcal{K}$ is quadratic in $\dot{q}$ in this model, the mass matrix is the velocity Hessian of the kinetic energy

$$\boldsymbol{M}(q) \;=\; \frac{\partial^2 \mathcal{K}}{\partial \dot{q}\,\partial \dot{q}}(q)$$

hence it is symmetric and, under standard independence and positivity assumptions on the inertia data, positive definite. The configuration dependence of $\boldsymbol{M}$ arises from the co-rotational frames; the off-diagonal blocks encode the inertial coupling between rigid angles and flexible degrees of freedom.

The term $\boldsymbol{H}(q,\dot{q})$ collects all contributions that do not multiply accelerations. A useful interpretation splits it as

$$\boldsymbol{H}(q,\dot{q}) \;=\; \boldsymbol{C}(q,\dot{q}) \;+\; \boldsymbol{g}(q) \;+\; \boldsymbol{k}_{int}(q),$$

where $\boldsymbol{C}$ contains convective, Coriolis/centrifugal, and gyroscopic effects generated by the q-dependence of $\partial\mathcal{K}/\partial\dot{q}$ (and vanishes at $\dot{q} = \boldsymbol{0}$), $\boldsymbol{g} = \partial\mathcal{U}_g/\partial q$ is the gravity contribution, and $\boldsymbol{k}_{int} = \partial\mathcal{U}_{\text{strain}}/\partial q$ is the internal elastic force derived from the strain energy. Equivalently,

$$\boldsymbol{H}(q,\dot{q}) \;=\; \underbrace{\left(\frac{\partial\mathcal{K}}{\partial q} - \frac{d}{dt}\frac{\partial\mathcal{K}}{\partial\dot{q}}\right)}_{\text{velocity-dependent and geometric terms}} \;+\; \underbrace{\frac{\partial\mathcal{U}}{\partial q}(q)}_{\text{gravity + elasticity}} .$$

Implicit time integration and Newton iterations require the consistent linearizations of $\boldsymbol{H}$ with respect to the state,

$$\boldsymbol{K}_t(q,\dot{q}) \;=\; \frac{\partial\boldsymbol{H}}{\partial q}(q,\dot{q}), \qquad \boldsymbol{C}_t(q,\dot{q}) \;=\; \frac{\partial\boldsymbol{H}}{\partial\dot{q}}(q,\dot{q}), \tag{5.31}$$

which are exact Jacobians (not ad-hoc approximations). They assemble the iteration matrix used by Newton's method and generalized-α; for the latter, the effective tangent typically appears as $\beta_p\,\boldsymbol{M} + \gamma_p\,\boldsymbol{C}_t + \boldsymbol{K}_t$. Retaining consistency in these derivatives preserves the expected quadratic convergence and the stability properties of the integrator.

When holonomic constraints are present, the form (5.30) becomes

$$\boldsymbol{M}(q)\,\ddot{q} \;+\; \boldsymbol{H}(q,\dot{q}) \;+\; \boldsymbol{J}_c(q)^{\top}\lambda \;=\; \tau, \qquad \boldsymbol{\phi}(q) = \boldsymbol{0},$$

where $\boldsymbol{J}_c(q) = \partial\boldsymbol{\phi}/\partial q$ is the constraint Jacobian and λ are Lagrange multipliers. In practice, the same $\boldsymbol{M}$, $\boldsymbol{H}$, $\boldsymbol{K}_t$, and $\boldsymbol{C}_t$ defined above are used within the constrained solver and in the projection steps that enforce the loop closure at position and velocity levels.

5.9 Element matrices and global assembly

This section summarizes how the elemental stiffness and mass matrices are formed in the co-rotational Timoshenko setting and how they are mapped and assembled into the global operators. The discussion follows standard finite-element practice for beams [13, 14, 15].

Local element operators (co-rotational frame)

Consider a two-node Timoshenko beam element of current chord length Δx expressed in its co-rotational *local* frame $\{t, n\}$, with the nodal degrees of freedom ordered as

$$\mathbf{q}_e^{\mathrm{loc}} = \begin{bmatrix} u_j & v_j & \theta_j & u_k & v_k & \theta_k \end{bmatrix}^\top,$$

where u is axial, v transverse, and θ the section rotation about the out-of-plane axis. The local (material) stiffness splits into an axial part and a bending–shear part,

$$\mathbf{k}_e^{\mathrm{loc}} \;=\; \frac{EA}{\Delta x}\,\mathbf{k}_{\mathrm{ax}} \;+\; \frac{EI}{\Delta x^3}\,\frac{1}{1+\phi}\,\mathbf{k}_{\mathrm{fl}}, \qquad \phi \;=\; \frac{12\,EI}{\kappa\,G\,A\,\Delta x^2}, \tag{5.32}$$

where ϕ is the usual shear-flexibility index; $\phi \to 0$ recovers the Euler–Bernoulli limit, while finite ϕ introduces Timoshenko shear compliance with correction factor κ. The consistent element mass matrix adds translational and rotary inertia contributions,

$$\mathbf{m}_e^{\mathrm{loc}} \;=\; \rho A\,\Delta x\,\mathbf{m}_{\mathrm{tr}} \;+\; \rho I\,\Delta x\,\mathbf{m}_{\mathrm{rot}}, \tag{5.33}$$

whose 6×6 blocks $(\mathbf{k}_{\mathrm{ax}}, \mathbf{k}_{\mathrm{fl}}, \mathbf{m}_{\mathrm{tr}}, \mathbf{m}_{\mathrm{rot}})$ are the canonical arrays for the two-node Timoshenko element; their explicit expressions are here reported (see also [13, 14]). Let $L_e = \Delta x$ denote the element length and $\phi = 12EI/(\kappa GAL_e^2)$. The axial part is

$$\boldsymbol{k}_{\mathrm{ax}} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The bending–shear part is

$$\boldsymbol{k}_{\mathrm{fl}} = \begin{bmatrix} 0 & 12 & 6L & 0 & -12 & 6L \\ 12 & 12\phi & 6L\phi & -12 & -12\phi & 6L\phi \\ 6L & 6L\phi & (4+\phi)L^2 & -6L & -6L\phi & (2-\phi)L^2 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ -12 & -12\phi & -6L\phi & 12 & 12\phi & -6L\phi \\ 6L & 6L\phi & (2-\phi)L^2 & -6L & -6L\phi & (4+\phi)L^2 \end{bmatrix}.$$

The consistent translational mass block is

$$\boldsymbol{m}_{\text{tr}} = \frac{1}{420}\begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix}.$$

The rotary inertia block acting on the θ-DOFs is

$$\boldsymbol{m}_{\text{rot}} = \frac{1}{6}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix}.$$

The full local matrices are then

$$\boldsymbol{k}_e^{\text{loc}} = \frac{EA}{L}\,\boldsymbol{k}_{\text{ax}} + \frac{EI}{L^3}\,\frac{1}{1+\phi}\,\boldsymbol{k}_{\text{fl}}, \qquad \boldsymbol{m}_e^{\text{loc}} = \rho AL\,\boldsymbol{m}_{\text{tr}} + \rho IL\,\boldsymbol{m}_{\text{rot}}.$$

In a co-rotational formulation the only operation needed to express the element operators in global coordinates is a rigid rotation consistent with the current element chord. Let the element direction be ψ_e, the rotation matrix be R_e:

$$c_e = \cos\psi_e, \qquad s_e = \sin\psi_e, \qquad \mathbf{R}_e = \begin{bmatrix} c_e & s_e & 0 \\ -s_e & c_e & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The 6×6 transformation acting on the element nodal vector is the block diagonal

$$\mathbf{T}_e \;=\; \operatorname{blkdiag}(\mathbf{R}_e, \mathbf{R}_e) \;=\; \begin{bmatrix} c_e & s_e & 0 & 0 & 0 & 0 \\ -s_e & c_e & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_e & s_e & 0 \\ 0 & 0 & 0 & -s_e & c_e & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.34}$$

With this mapping, the global element matrices are obtained by congruence:

$$\mathbf{k}_e \;=\; \mathbf{T}_e^{\top}\mathbf{k}_e^{\text{loc}}\mathbf{T}_e, \qquad \mathbf{m}_e \;=\; \mathbf{T}_e^{\top}\mathbf{m}_e^{\text{loc}}\mathbf{T}_e. \tag{5.35}$$

No further geometric terms are introduced at this stage: the co-rotational approach preserves the canonical local operators and accounts for large rigid motions through $\mathbf{T}_e$ only [16, 15].

Boolean assembly into global operators

Let $\mathbf{A}_e$ be the Boolean (scatter) matrix that places the six element DOFs into their positions in the global vector of active coordinates. Its construction follows the active-DOF numbering, so that locked DOFs simply do not appear in $\mathbf{A}_e$. The global stiffness and mass matrices are assembled as

$$\mathbf{K} \;=\; \sum_{e=1}^{N_e} \mathbf{A}_e^\top \mathbf{k}_e \mathbf{A}_e, \qquad \mathbf{M} \;=\; \sum_{e=1}^{N_e} \mathbf{A}_e^\top \mathbf{m}_e \mathbf{A}_e. \tag{5.36}$$

By construction $\mathbf{K}$ and $\mathbf{M}$ are symmetric. The mass matrix $\mathbf{M}$ is positive definite on the subspace of independent coordinates; the stiffness $\mathbf{K}$ is positive semidefinite if rigid-body modes are unconstrained and becomes positive definite once boundary conditions and loop constraints are enforced [17, 21].

A practical and robust diagnostic is to compare the assembled matrices for an inclined mesh against a reference assembly built for a horizontal mesh of the same segmentation and properties, related by a global rotation. Denote by $(\mathbf{K}_{\text{incl}}, \mathbf{M}_{\text{incl}})$ the operators assembled with element-wise rotations as in (5.35), and by $(\mathbf{K}_x, \mathbf{M}_x)$ those assembled with all elements aligned with the global x-axis. Let $\mathbf{T}$ be the global block-diagonal rotation that applies $\mathbf{R}_e$ to each node that carries the full triplet (u, v, θ). Then one expects the congruence relations

$$\mathbf{K}_{\text{incl}} \;\approx\; \mathbf{T}^\top \mathbf{K}_x \mathbf{T}, \qquad \mathbf{M}_{\text{incl}} \;\approx\; \mathbf{T}^\top \mathbf{M}_x \mathbf{T},$$

and the relative Frobenius norms

$$\frac{\|\mathbf{K}_{\text{incl}} - \mathbf{T}^\top \mathbf{K}_x \mathbf{T}\|_F}{\|\mathbf{K}_x\|_F} \ll 1, \qquad \frac{\|\mathbf{M}_{\text{incl}} - \mathbf{T}^\top \mathbf{M}_x \mathbf{T}\|_F}{\|\mathbf{M}_x\|_F} \ll 1, \tag{5.37}$$

which effectively detect mistakes in local-to-global rotations, DOF ordering, or Boolean scatter; see, [24, 25]. This check mirrors the verification used in the implementation and is recommended as a regression test whenever the mesh description or numbering changes.

5.10 Loop-closure and constrained equations

The mechanism has a single holonomic closure that enforces coincidence between the flexible beam tip and the tip of the upper rigid link. Let $\boldsymbol{p}_B(q) \in \mathbb{R}^2$ denote the current global position of the beam tip, obtained from the co-rotational geometry (Section 5.9 and the routine `current_geom_from_q`), and let

$$\boldsymbol{p}_3(q_3) \;=\; \begin{bmatrix} L_3 \cos q_3 \\ L_0 + L_3 \sin q_3 \end{bmatrix}$$

be the current tip of link 3, whose base is fixed at $C = [0, L_0]^\top$. The vector-valued constraint map

$$\boldsymbol{\phi}(q) \;=\; \boldsymbol{p}_B(q) \;-\; \boldsymbol{p}_3(q_3) \;\in \mathbb{R}^2 \tag{5.38}$$

measures the translational mismatch in global coordinates. Imposing $\boldsymbol{\phi}(q) = \mathbf{0}$ closes the loop by making the two tips coincide.

The constraint Jacobian is

$$\boldsymbol{J}_c(q) \;=\; \frac{\partial \boldsymbol{\phi}}{\partial q}(q) \;\in\; \mathbb{R}^{2\times n}. \tag{5.39}$$

Rank $\boldsymbol{J}_c(q) = 2$ in the configurations of interest, i.e., there are no alignment singularities that make the two scalar constraints locally dependent. Under this rank condition, the local constraint collector $\{q : \boldsymbol{\phi}(q) = \mathbf{0}\}$ is smooth and of codimension 2.

Introducing Lagrange multipliers $\lambda \in \mathbb{R}^2$ with the physical meaning of constraint reaction forces at the tip, the unconstrained standard form

$$\mathbf{M}(q)\,\ddot{q} + \mathbf{H}(q, \dot{q}) = \tau \tag{5.40}$$

is augmented to the semi-explicit index-2 DAE

$$\begin{cases} \mathbf{M}(q)\,\ddot{q} \;+\; \mathbf{H}(q, \dot{q}) \;+\; \mathbf{J}_c(q)^\top \lambda \;=\; \tau, \\ \boldsymbol{\phi}(q) \;=\; \mathbf{0}, \end{cases} \tag{5.41}$$

where $\mathbf{J}_c(q) = \partial\boldsymbol{\phi}/\partial q$. This form enforces the closure at the position level and ensures that the reaction forces do no work along admissible virtual displacements.

5.10.1 Acceleration-level stabilization (Baumgarte).

During time integration the constraint $\boldsymbol{\phi}(q) = \mathbf{0}$ can slowly drift away from zero because of numerical errors. A simple way to limit this drift is to add the following acceleration-level correction

$$\mathbf{J}_c(q)\,\ddot{q} \;+\; 2\,\zeta\,\omega_B\,\mathbf{J}_c(q)\,\dot{q} \;+\; \omega_B^2\,\boldsymbol{\phi}(q) \;=\; \mathbf{0}, \tag{5.42}$$

where: $\boldsymbol{\phi}(q) \in \mathbb{R}^2$ is the two-component constraint function (it should be zero), $\mathbf{J}_c(q) = \partial\boldsymbol{\phi}/\partial q$ is its Jacobian, q are the generalized coordinates, $\dot{q}$ and $\ddot{q}$ are velocity and acceleration, $\zeta \in [0,1]$ is a dimensionless damping ratio, and $\omega_B > 0$ is a stabilization angular frequency (in s^{-1}). The last term, $\omega_B^2\,\boldsymbol{\phi}(q)$, pulls the positions back toward zero constraint error; the middle term, $2\,\zeta\,\omega_B\,\mathbf{J}_c(q)\dot{q}$, damps

the constraint-related velocities; the first term ensures the correction acts at the acceleration level. In the code, this appears with coefficients $2\,z_B\,w_B$ and w_B^2, so that $\zeta = z_B$ and $\omega_B = w_B$.
Choosing the two parameters follows common practice in constrained multibody dynamics. It is convenient to tie the frequency to the time step Δt so that the correction acts on the same time scale as the integrator, for example

$$\omega_B \;=\; \frac{\kappa}{\Delta t}, \qquad \kappa \in [10,\,30],$$

and to use a well-damped setting for ζ, e.g. $\zeta \approx 0.5$–0.9. These ranges are widely used and documented in the literature on constrained dynamics and DAEs; see [26, 21, 10, 22]. With this choice, the stabilization acts as a critically sized "virtual spring–damper" on the constraints: too small ω_B allows drift to accumulate, whereas too large ω_B makes the algebraic block overly stiff and can degrade Newton convergence.

For completeness, many solvers add to (5.42) also a projection step after each corrector (positions: weighted least squares to enforce $\boldsymbol{\phi}(q) = \mathbf{0}$; velocities: orthogonal projection to enforce $\mathbf{J}_c(q)\,\dot{q} = \mathbf{0}$), which further reduces drift without requiring very large ω_B [9, 10].

After each nonlinear corrector step (or whenever the user chooses), the positions and velocities are corrected so that the loop-closure constraint is satisfied again. This procedure is usually called a projection step.
The correction for the positions is computed by solving a small weighted least-squares problem: it finds the smallest change in the coordinates that brings the constraint value $\boldsymbol{\phi}(q)$ back to zero, while keeping some degrees of freedom almost fixed if they are assigned a large weight:

$$\min_{\Delta q}\; \tfrac{1}{2}\,\Delta q^{\top}\mathbf{W}\,\Delta q \quad \text{subject to} \quad \boldsymbol{\phi}(q + \Delta q) = \mathbf{0},$$

with a positive diagonal weight $\mathbf{W} = \mathrm{diag}(w_i)$ that can be used to keep selected coordinates effectively fixed by setting large weights w_i. Linearizing $\boldsymbol{\phi}(q + \Delta q) \approx \boldsymbol{\phi}(q) + \mathbf{J}_c(q)\Delta q$ yields the closed-form correction

$$\Delta q \;=\; -\,\mathbf{W}^{-1}\,\mathbf{J}_c(q)^{\top}\left(\mathbf{J}_c(q)\,\mathbf{W}^{-1}\,\mathbf{J}_c(q)^{\top}\right)^{-1}\boldsymbol{\phi}(q), \tag{5.43}$$

so that $q \leftarrow q + \Delta q$ satisfies $\boldsymbol{\phi}(q) = \mathbf{0}$ up to linearization error. Velocities are then orthogonally projected onto the null space of the constraints,

$$\dot{q} \;\leftarrow\; \dot{q} \;-\; \mathbf{J}_c(q)^{\top}\Big(\mathbf{J}_c(q)\,\mathbf{J}_c(q)^{\top}\Big)^{-1}\mathbf{J}_c(q)\,\dot{q}, \tag{5.44}$$

which enforces $\mathbf{J}_c(q)\,\dot{q} = \mathbf{0}$. These two steps mirror the routine `project_loop` and are numerically robust as long as $\operatorname{rank}\mathbf{J}_c = 2$.

Within an implicit integrator (e.g., generalized-α), the nonlinear residual for the momentum equation and the stabilized constraints is linearized at each iteration. Denote by $\mathbf{S}_t$ the effective tangent arising from the scheme (a combination of $\mathbf{M}$, $\mathbf{C}_t$, and $\mathbf{K}_t$), and by $\mathbf{r}_m$ and $\mathbf{r}_c$ the current residuals of the momentum and constraint equations, respectively. The Newton step solves a saddle-point (KKT) system of the form

$$\begin{bmatrix} \mathbf{S}_t & \mathbf{J}_c^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_m \\ \mathbf{r}_c \end{bmatrix}, \qquad \mathbf{B} \equiv \beta_p\,\mathbf{J}_c + \gamma_p\,\widetilde{\mathbf{J}}_c, \tag{5.45}$$

where the blocks β_p, γ_p and $\widetilde{\mathbf{J}}_c$ depend on the chosen time integrator and on the stabilization (5.42). Eliminating Δq by a Schur complement on the 2×2 multiplier block provides a well-conditioned update for $\Delta\lambda$, after which Δq is recovered by a single back-substitution. This is the pattern implemented in the solver, with a mild diagonal regularization of $\mathbf{S}_t$ to improve numerical robustness.

5.10.2 Practical checks.

Regularly monitoring $\operatorname{rank}\mathbf{J}_c$, the norm of $\boldsymbol{\phi}(q)$ after projection, and the size of the correction Δq in (5.43) is good practice. Large projections or loss of rank often signal near-singular configurations (e.g., aligned bars) that require smaller time steps or alternative parameterizations.

The formulation above is standard in constrained multibody dynamics and differential–algebraic equations [9, 10].

```
function phi = loop_phi(prop, q, nf)
% loop_phi  Two closure constraints: beam tip B equals link-3 tip.
% Inputs:
%   prop : struct with P1 (2x1), L0, L1, L2, L3
%   q    : generalized coordinates [q1;q2;q3; qd]
%   nf   : (Ne+1)x3 numbering of active DOFs (u,v,theta)
% Output:
%   phi  : 2x1 mismatch pB - p3 (should be zero at closure)

q1 = q(1); q2 = q(2); q3 = q(3);
psi = q1 + q2;

% Link-1 tip (point A)
pA = prop.P1 + prop.L1 * [cos(q1); sin(q1)];

% Beam tip (point B): baseline reach + (u,v) tip in GLOBAL axes
pB = pA + prop.L2 * [cos(psi); sin(psi)];
```

```
i_tip = size(nf,1);
iu = nf(i_tip,1); iv = nf(i_tip,2);
if iu>0, pB = pB + [q(3+iu); 0]; end
if iv>0, pB = pB + [0; q(3+iv)]; end

% Link-3 tip
p3 = [ prop.L3*cos(q3);
       prop.L0 + prop.L3*sin(q3) ];

phi = pB - p3;  % 2x1
end


function Jc = loop_Jc(prop, q, nf)
% loop_Jc  Jacobian of the closure constraint phi(q).
% Inputs/Output as in loop_phi. Output: Jc is 2-by-n.

q1 = q(1); q2 = q(2); q3 = q(3);
psi = q1 + q2;
n   = numel(q);
Jc  = zeros(2,n);

% d pB / d q1 and d pB / d q2 (baseline contributions)
Jc(:,1) = [-prop.L1*sin(q1) - prop.L2*sin(psi);
            prop.L1*cos(q1) + prop.L2*cos(psi)];
Jc(:,2) = [-prop.L2*sin(psi);
            prop.L2*cos(psi)];

% d(-p3)/dq3
Jc(:,3) = [ +prop.L3*sin(q3);
            -prop.L3*cos(q3) ];

% Tip DOFs (u,v) mapped 1:1 in GLOBAL axes
i_tip = size(nf,1);
iu = nf(i_tip,1); iv = nf(i_tip,2);
if iu>0, Jc(:, 3+iu) = [1;0]; end
if iv>0, Jc(:, 3+iv) = [0;1]; end
end
```

Listing 5.3: Loop-closure maps: position constraint `phi(q)` and Jacobian `Jc(q)`.

5.11 Tendon model, endpoint Jacobians, and static pretension

A cable (tendon) is modeled between a point A near the tip of link 1 and the flexible–beam tip B (as in the PACOMA concept, cf. [1]). Let

$$\boldsymbol{r}(q) = \boldsymbol{p}_B(q) - \boldsymbol{p}_A(q), \qquad L(q) = \|\boldsymbol{r}(q)\|, \qquad \boldsymbol{e}(q) = \frac{\boldsymbol{r}(q)}{L(q)},$$

be, respectively, the relative position, its length, and the unit direction from A to B in global coordinates.

The endpoint positions depend on the generalized coordinates q. The beam tip $\boldsymbol{p}_B(q)$ comes from the co-rotational geometry used for the flexible limb (`current_geom_from_q`), so it depends on (q_1, q_2) and on the active nodal DOFs of the last beam node (u_{N+1}, v_{N+1}), as well as weakly on nearby elastic DOFs through the last-element orientation. The upper-link tip $\boldsymbol{p}_3(q_3)$ is handled by loop-closure and does not enter the tendon directly.
The attachment on link 1 is placed at

$$\boldsymbol{p}_A(q) = \underbrace{\begin{bmatrix} 0.150 \\ 0 \end{bmatrix} + L_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix}}_{\text{tip of link 1}}$$

so its Jacobian has a single nonzero column (with respect to q_1):

$$\boldsymbol{J}_A(q) \equiv \frac{\partial \boldsymbol{p}_A}{\partial q}(q)\ , \qquad \boldsymbol{J}_A(:,1) = \begin{bmatrix} -L_1 \sin q_1 \\ L_1 \cos q_1 \end{bmatrix}, \quad \boldsymbol{J}_A(:,i) = \boldsymbol{0} \ \ (i \neq 1).$$

The beam-tip Jacobian

$$\boldsymbol{J}_B(q) \equiv \frac{\partial \boldsymbol{p}_B}{\partial q}(q)$$

includes: (i) the rigid contribution from (q_1, q_2) along the baseline direction, and (ii) the mapping of the tip DOFs u_{N+1}, v_{N+1} in the current local frame. In practice, the code fills the two nonzero columns for u_{N+1} and v_{N+1} with $[1\ 0]^\top$ and $[0\ 1]^\top$ in global axes, and adds the baseline part in the first two angle columns. The relative Jacobian

$$\boldsymbol{J}_{\text{rel}}(q) = \boldsymbol{J}_B(q) - \boldsymbol{J}_A(q)$$

maps generalized velocities to the relative endpoint velocity: $\dot{\boldsymbol{r}} = \boldsymbol{J}_{\text{rel}}(q)\,\dot{q}$.

5.11.1 Spring–damper along the cable line.

A linear spring–damper acts only along the current cable direction $\boldsymbol{e}(q)$. The scalar cable force is

$$F(q,\dot{q}) \;=\; k\Big(L(q)-L_0\Big) \;+\; c \quad \underbrace{\boldsymbol{e}(q)^\top \boldsymbol{J}_{\mathrm{rel}}(q)\dot{q}}_{\text{relative speed along the cable}} \quad , \tag{5.46}$$

and the corresponding global force at the endpoints is $\boldsymbol{f} = F\,\boldsymbol{e}$ (tension is positive). By virtual work, the generalized force contributed by the tendon is obtained by the standard Jacobian–transpose mapping [27, 28]:

$$\boldsymbol{Q}_{\mathrm{ten}}(q,\dot{q}) \;=\; \boldsymbol{J}_{\mathrm{rel}}(q)^\top \boldsymbol{f} \;=\; \boldsymbol{J}_{\mathrm{rel}}(q)^\top \boldsymbol{e}(q)\,F(q,\dot{q}). \tag{5.47}$$

For Newton iterations it is useful to include the (rank–2) elastic tangent of the spring along the current direction,

$$\boldsymbol{K}_{\mathrm{ten}}(q) \;=\; \boldsymbol{J}_{\mathrm{rel}}(q)^\top \Big(k\,\boldsymbol{e}(q)\boldsymbol{e}(q)^\top\Big)\,\boldsymbol{J}_{\mathrm{rel}}(q), \tag{5.48}$$

which follows from linearizing the spring part of (5.46).

5.11.2 Static pretension problem.

A pretensioned reference shape is found by solving a static equilibrium with the loop closed. One seeks $q^\star$ such that internal elastic forces and gravity balance the tendon, while the tip B still coincides with the upper-link tip. Using the notation of Section 5.9,

$$\underbrace{\boldsymbol{H}(q,\boldsymbol{0})}_{\text{gravity + elastic at rest}} \;+\; \underbrace{\boldsymbol{Q}_{\mathrm{ten}}(q,\boldsymbol{0})}_{\text{spring tension}} \;=\; \boldsymbol{0}, \qquad \boldsymbol{\phi}(q)=\boldsymbol{0}. \tag{5.49}$$

The solution proceeds with Newton's method on the residual $\boldsymbol{R}(q) = \boldsymbol{H}(q,\boldsymbol{0}) + \boldsymbol{Q}_{\mathrm{ten}}(q,\boldsymbol{0})$, using the consistent tangent $\boldsymbol{K}_t(q,\boldsymbol{0}) + \boldsymbol{K}_{\mathrm{ten}}(q)$ inside each iteration. It is often convenient to hold q_3 at a target value (e.g., $q_3 = 0$) as a hard constraint in the linear system; q_1 can also be fixed if desired. After each Newton step, a short projection enforces the loop closure $\boldsymbol{\phi}(q) = \boldsymbol{0}$ (section 5.10).

As an illustration of the target operating posture produced by (5.49), Fig. 5.4 shows a rendering from MSC Adams in which a spring–damper along the tendon pulls the flexible limb outward, producing a stable pretensioned shape. The image is provided only to clarify geometry and intended deformation. On MSC Adams, the pretensioned configuration requires 10 seconds to stabilize, it is not possible to get a static pretension starting from the rest position.

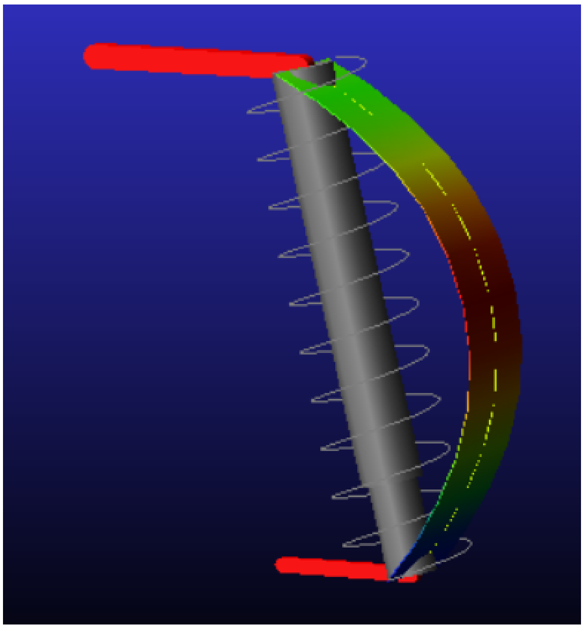


Figure 5.4: Qualitative rendering of the pre-tensioned leg (MSC Adams). The cable runs from a point near the tip of link 1 to the flexible–beam tip, generating an outward bend under pretension. Colors indicate bending intensity; red bars are the rigid links.

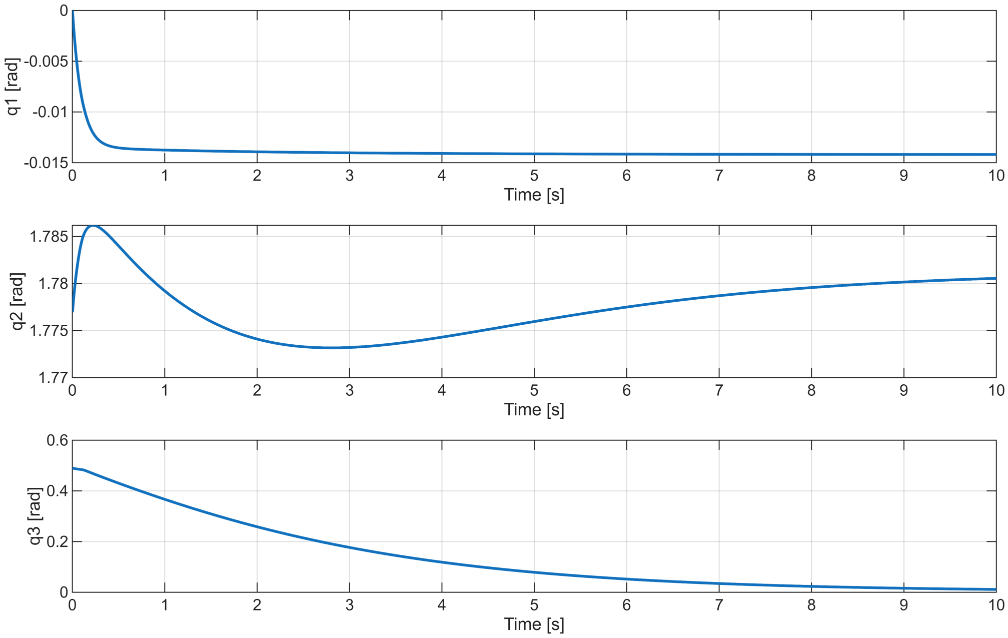


Figure 5.5: Angles q_1, q_2 and q_3 during the pretensioning transient.

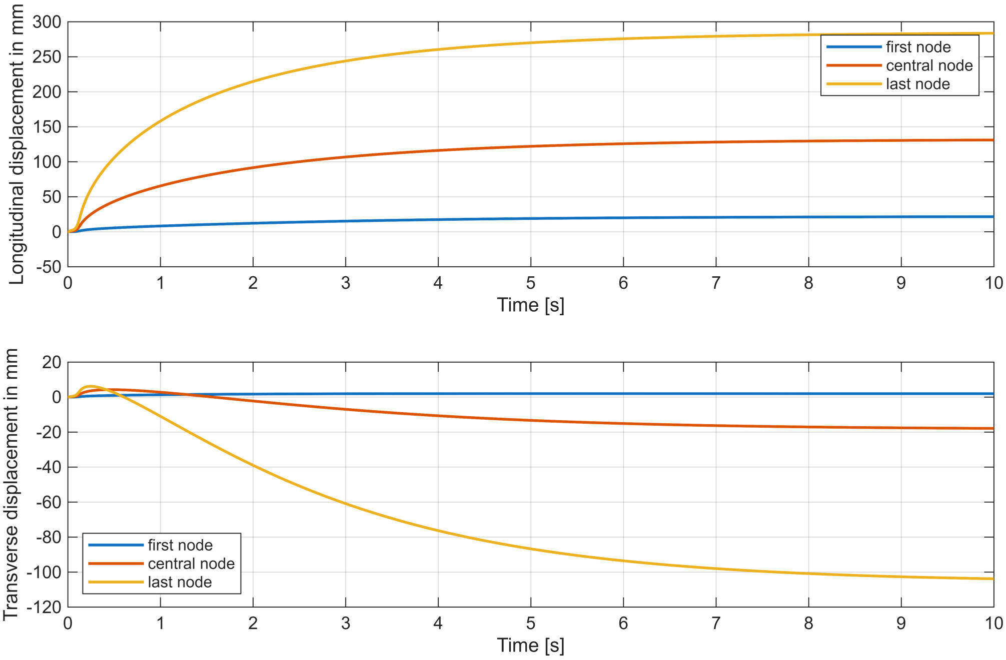


Figure 5.6: The displacement of the three internal nodes of the flexible beam (when it is divided into 4 elements) during the pretensioning transient.

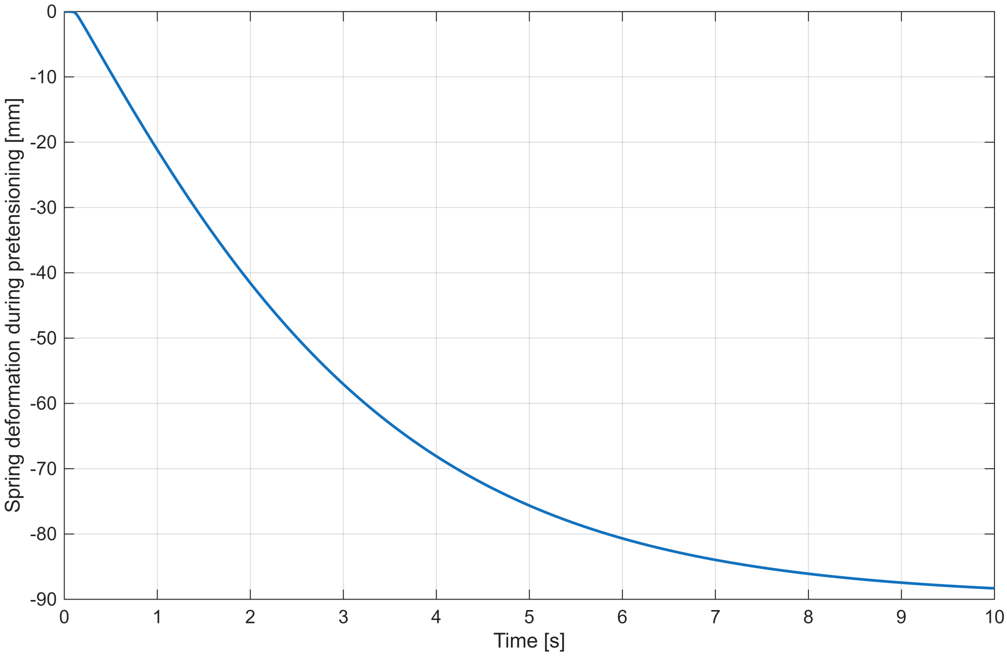


Figure 5.7: Spring deformation during the pretensioning transient.

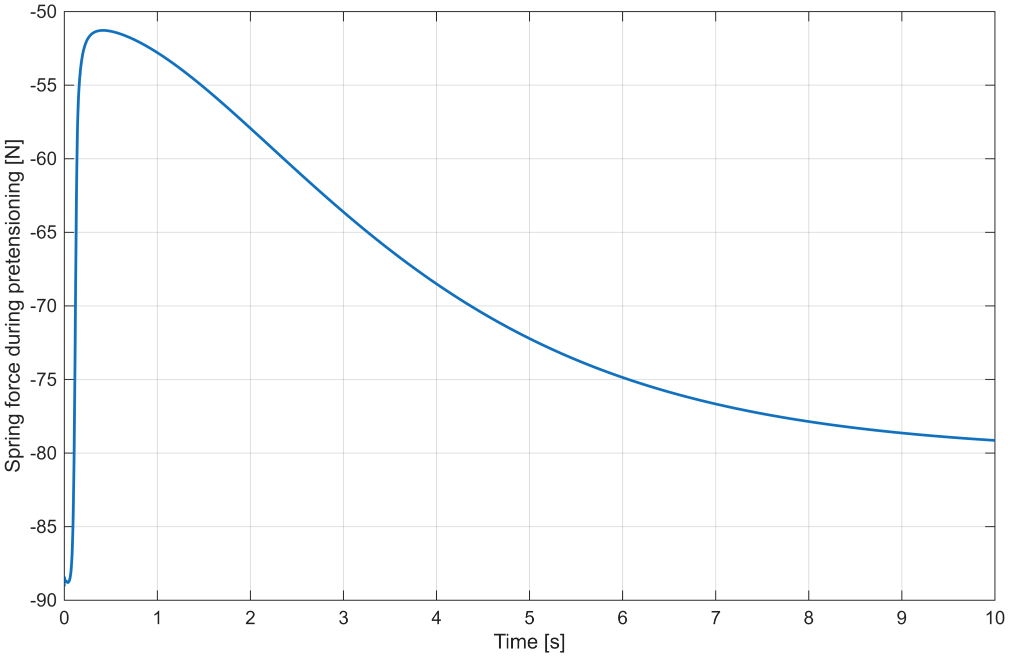


Figure 5.8: Spring force during the pretensioning transient.

Figure 5.5 shows the behavior of the three rigid angles during the pretensioning transient. This transient is the movement of the model from a state like the one represented in Figure 5.2 to the final pretensioned model shown in Figure 5.4, which is the starting point for dynamic simulations. Figure 5.6 shows the displacements during the pretension transient of the three internal nodes of a 4-elements Timoshenko beam, where the "first node" indicates the closest internal node to the lower rigid link, and the "last node" the closest one to the upper rigid link.

Figures 5.7 and 5.8 present, respectively, the evolution of the deformation and the force developed by the spring–damper during the transient pretension phase carried out in MSC Adams. This stage of the simulation is crucial, as it defines the initial mechanical state from which the subsequent dynamic response of the system originates. To impose the desired prestress, an initial compressive load of 89 N was applied to the spring. This specific value was determined through preliminary analyses aimed at identifying the load level capable of guiding the flexible element toward its optimal pretensioned configuration, while at the same time minimizing undesired oscillations and ensuring a stable convergence of the mechanism toward the targeted initial state. The selected pretension force therefore represents the most effective compromise between achieving the required deformation and maintaining numerical and mechanical stability during the transient process.

A practical trick improve robustness: a very small geometric imperfection added to a middle interior u-DOFs to make the beam choose a definite bending side under tension, instead of converging to a symmetric but unstable straight state.

```
function [Qten, Kten, F, ehat, Llen] = tendon_loads(q, dq, prop,
    nf, k, c, L0)
% tendon_loads  Spring-damper along the A->B line (positive
    tension).
% Inputs:
%   q,dq : state
%   prop : struct with P1, L0, L1, L2, L3
%   nf   : numbering of active DOFs (u,v,theta)
%   k,c  : stiffness [N/m] and damping [N*s/m]
%   L0   : rest length [m]
% Outputs:
%   Qten : generalized force (n x 1)
%   Kten : elastic tangent (n x n), rank 2
%   F    : scalar cable tension (N)
%   ehat : unit direction 2x1 from A to B
%   Llen : current cable length (m)

% Endpoints
q1 = q(1); q2 = q(2);
pA = prop.P1 + prop.L1 * [cos(q1); sin(q1)];

psi = q1 + q2;
pB = pA + prop.L2 * [cos(psi); sin(psi)];

% Tip (u,v) in GLOBAL
i_tip = size(nf,1);
iu = nf(i_tip,1); iv = nf(i_tip,2);
if iu>0, pB = pB + [q(3+iu); 0]; end
if iv>0, pB = pB + [0; q(3+iv)]; end

% Relative vector, length, direction
r    = pB - pA;
Llen = max(norm(r), 1e-12);
ehat = r / Llen;

% Jacobians of endpoints (2 x n)
n   = numel(q);
J_A = zeros(2,n);
J_A(:,1) = [-prop.L1*sin(q1);  prop.L1*cos(q1)];   % only q1

J_B = J_A;                                        % includes q1 part
J_B(:,2) = [-prop.L2*sin(psi);  prop.L2*cos(psi)];  % q2
if iu>0, J_B(:, 3+iu) = [1;0]; end                  % u_tip
if iv>0, J_B(:, 3+iv) = [0;1]; end                  % v_tip

```

```
Jrel = J_B - J_A;                                    % dq -> rdot

% Scalar force of spring-damper along ehat
vrel = ehat.' * (Jrel * dq);                   % relative speed
F    = k*(Llen - L0) + c*vrel;                      % tension
fvec = ehat * F;                            % 2x1 nodal force

% Map to generalized forces and elastic tangent
Qten = Jrel.' * fvec;
Kten = Jrel.' * (k * (ehat*ehat.')) * Jrel;
end
```

Listing 5.4: Tendon generalized force `Qten` and elastic tangent `Kten`.

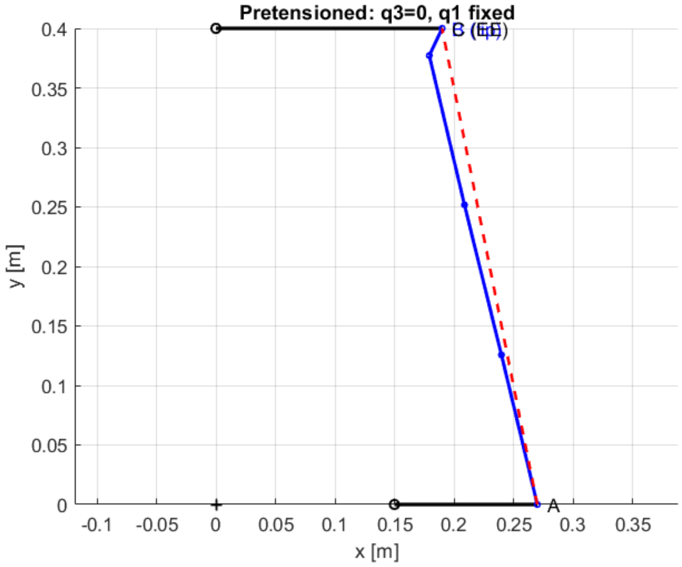


Figure 5.9: Static pretension configuration predicted by the current co-rotational Timoshenko model.

Figure 5.9 shows the static pretension configuration obtained by solving the nonlinear equilibrium problem in Eq. (5.49) with the present co-rotational Timoshenko formulation. Instead of producing a smooth outward bend qualitatively similar to the MSC Adams reference in Fig. 5.4, the finite-element model keeps the beam almost straight and concentrates the curvature in a single “kink” at the tip. This is inconsistent with the expected post-buckling shape of a slender, initially straight leaf spring under end–to–end compression and tendon pretension.

To better understand this behavior, a reduced four-element test mesh was used for diagnostics. After convergence of the static solve, the element lengths and axial strains were post-processed directly from the nodal coordinates. The resulting data are summarized in Table 5.3.

Element	L_e [m]	Axial strain ε [-]
1	0.1308	$4.65 \cdot 10^{-2}$
2	0.1293	$3.42 \cdot 10^{-2}$
3	0.1286	$2.87 \cdot 10^{-2}$
4	0.0246	$-8.03 \cdot 10^{-1}$

Table 5.3: Element lengths and axial strains in the static pretension solution of a four-element test mesh. The first three elements are in mild tension, while almost the entire compression is localized in the last element, which exhibits an unrealistically large negative strain.

The first three elements are slightly elongated and carry only moderate positive axial strain, whereas the last element becomes extremely short, with an axial strain of about -80%. From a mechanical viewpoint this equilibrium is not acceptable for the titanium leaf spring considered here: the compression is numerically concentrated in a single element instead of being smoothly distributed along the span, and the local strain level is far beyond the range in which the adopted small-strain, von Kármán co-rotational formulation is meant to operate [16, 15].
The mismatch between Fig. 5.9 and the target shape in Fig. 5.4 is therefore interpreted as a *limitation of the current modeling and discretization choices.* In particular:

- the present co-rotational Timoshenko model with von Kármán axial strain assumes small strains in the element-local frame; the highly compressed last element in Table 5.3 violates this assumption and pushes the formulation outside its intended validity range;
- the static pretension problem is strongly nonlinear (the tendon force exceeds the first Euler buckling load of the beam), and with a coarse mesh the Newton iterations tend to converge to a localized post-buckling branch where one element carries most of the compression, a behavior that is well known in classical FEM post-buckling analyses when the discretization is too coarse or the regularization too weak [16, 15];
- the pretension is applied only through an end–to–end tendon; no additional distributed features (e.g. imperfections, auxiliary springs, or stiffness regularization) are present to penalize a local collapse of the tip element relative to the rest of the beam.

Several remedies could be considered in future work to obtain a pretensioned shape that is quantitatively closer to the reference:

- increasing the number of finite elements used in the static pretension stage, so that the curvature can be distributed over many elements instead of being forced to localize in one, but this choice would lead to an exponential increase in computational costs;
- solving the static problem with a continuation or arc-length strategy, rather than with simple load control, to follow the smooth post-buckling branch and avoid snap-through to a strongly localized equilibrium;
- introducing a small geometric imperfection or additional regularization (for example a very mild penalty on abrupt curvature changes near the tip) to discourage the collapse of a single element and promote smoother shapes;
- re-tuning the tendon parameters (stiffness and imposed shortening) so that the operating pretension level remains closer to the first buckling threshold predicted by the linearized beam model.

In this thesis the pretensioned configuration actually used as a reference for the leg motion is still taken from the high-fidelity MSC Adams model (Fig. 5.4), while the MATLAB configuration in Fig. 5.9 is documented as a known limitation of the current co-rotational implementation and as a starting point for future improvements of the flexible-leg model.

5.12 Time integration: generalized-α scheme with constraints

Time advancement is carried out with the second-order generalized-α method for structural dynamics. The method is implicit, unconditionally stable under standard parameter choices, and offers controllable damping of the very high-frequency content while keeping second-order accuracy at the time step of interest [6, 13, 17].

Parameters and algorithmic states

A single user choice, the target spectral radius at infinity $\rho_\infty \in [0,1]$, sets the high-frequency numerical damping: $\rho_\infty = 1$ gives no artificial dissipation (Newmark-like), while $\rho_\infty \to 0$ increases decay of spurious high modes. Given ρ_∞, the generalized-α parameters are taken as

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \quad ; \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} \quad ; \quad \gamma = \tfrac{1}{2} - \alpha_m + \alpha_f \quad ; \quad \beta = \tfrac{1}{4}\left(1 - \alpha_m + \alpha_f\right)^2 \tag{5.50}$$

which ensure second-order accuracy and the desired decay at infinity [6].
At time $t_{n+1} = t_n + \Delta t$, residuals and tangents are evaluated at the algorithmic (weighted) states

$$\begin{aligned} q_{\text{eff}}^{n+1} &= (1-\alpha_f)\, q^{n+1} + \alpha_f\, q^n \\ \dot{q}_{\text{eff}}^{n+1} &= (1-\alpha_f)\, \dot{q}^{n+1} + \alpha_f\, \dot{q}^n \\ \ddot{q}_{\text{eff}}^{n+1} &= (1-\alpha_m)\, \ddot{q}^{n+1} + \alpha_m\, \ddot{q}^n \end{aligned} \tag{5.51}$$

Predictor formulas of Newmark type supply initial guesses for $(q^{n+1}, \dot{q}^{n+1}, \ddot{q}^{n+1})$, which are then corrected by Newton iterations.

Effective iteration matrix

Let $\mathbf{M}(q)$ be the generalized mass matrix and let $\mathbf{H}(q, \dot{q})$ collect all non-inertial terms (elastic internal forces, gravity, and velocity-dependent effects) so that the unconstrained form is $\mathbf{M}(q)\ddot{q} + \mathbf{H}(q, \dot{q}) = \tau$ (see the previous section). Linearizing the discrete residual at the states (5.51) yields the effective matrix acting on the configuration increment Δq:

$$\mathbf{S}_t \;=\; \beta_p\, \mathbf{M} \;+\; \gamma_p\, \mathbf{C}_{\text{tot}} \;+\; \mathbf{K}_t \quad ; \quad \beta_p = \frac{1-\alpha_m}{\beta(1-\alpha_f)\,\Delta t^2} \quad ; \quad \gamma_p = \frac{\gamma}{\beta(1-\alpha_f)\,\Delta t} \tag{5.52}$$

where $\mathbf{K}_t = \partial \mathbf{H}/\partial q$ and $\mathbf{C}_t = \partial \mathbf{H}/\partial \dot{q}$ are the consistent tangents. If structural (Rayleigh) damping is used, it is added as

$$\mathbf{C}_{\text{tot}} \;=\; \mathbf{C}_t \;+\; \mathbf{C}_{\text{Ray}}, \qquad \mathbf{C}_{\text{Ray}} = \alpha_R\, \mathbf{M} + \beta_R\, \mathbf{K}_t, \tag{5.53}$$

with α_R, β_R explained in section 5.13 [17, 13].

Coupling with loop-closure constraints

Loop-closure is enforced by Lagrange multipliers $\lambda \in \mathbb{R}^2$ and, optionally, by an acceleration-level stabilization (Baumgarte) to limit drift; denote $\boldsymbol{\phi}(q) = \mathbf{0}$ and $\mathbf{J}_c(q) = \partial \boldsymbol{\phi}/\partial q$. Each Newton corrector solves the coupled system

$$\begin{bmatrix} \mathbf{S}_t & \mathbf{J}_c^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_{\text{dyn}} \\ -\mathbf{r}_c \end{bmatrix}. \tag{5.54}$$

Here $\mathbf{R}_{\text{dyn}}$ is the dynamic residual assembled with (5.51). The lower block depends on the chosen constraint level: for a pure position correction, $\mathbf{r}_c = \boldsymbol{\phi}(q)$ and $\mathbf{B} = \mathbf{J}_c(q)$; for acceleration-level stabilization, $\mathbf{r}_c = \mathbf{J}_c\, \ddot{q} + 2\zeta_B \omega_B\, \mathbf{J}_c\, \dot{q} + \omega_B^2\, \boldsymbol{\phi}(q)$ and $\mathbf{B}$ is the corresponding Jacobian with respect to q (see the dedicated section on stabilization). Eliminating Δq gives a 2×2 Schur complement for $\Delta \lambda$,

$$\mathbf{S}_\lambda\, \Delta \lambda \;=\; -\mathbf{r}_c + \mathbf{B}\, \mathbf{S}_t^{-1}\, \mathbf{R}_{\text{dyn}} \quad ; \quad \mathbf{S}_\lambda \;=\; -\,\mathbf{B}\, \mathbf{S}_t^{-1}\, \mathbf{J}_c^\top. \tag{5.55}$$

After convergence at each step, positions and velocities are corrected to remove any small residual violation of the loop: the position correction solves a weighted least-squares problem that finds the smallest change Δq bringing $\boldsymbol{\phi}(q + \Delta q)$ back to zero while keeping selected coordinates effectively fixed via large weights; the velocity correction enforces $\mathbf{J}_c(q)\,\dot{q} = \mathbf{0}$. These simple projections are standard for index-2 constrained mechanical systems and reduce the need for very strong stabilization [9, 10].

Practical settings

The scheme is second-order accurate with (5.50). Values such as $\rho_\infty \simeq 0.5$ provide a good compromise between damping of high-frequency numerical content and preservation of low-frequency motion; $\rho_\infty \simeq 0.7$–0.9 reduces dissipation when cleaner vibration content is desired. These ranges are widely used in structural dynamics practice [6, 13].

```matlab
function [dq_step, dlam] = ga_newton_step(q, dq, ddq, lam, pars, model)
% ga_newton_step  Solve the coupled linear system for (Delta q, Delta lambda)
% produced by the generalized-alpha corrector with constraints.
%
% Inputs:
%   q,dq,ddq : current iterates
%   lam      : current Lagrange multipliers
%   pars     : struct with fields:
%              betap, gammap, tau, use_baumgarte (logical), zB, wB
%   model    : struct with function handles:
%              [M,H,Kt,Ct,Cray] = model.operators(q,dq)
%              phi = model.loop_phi(q)
%              Jc  = model.loop_Jc(q)
%
% Outputs:
%   dq_step  : configuration correction (n x 1)
%   dlam     : multiplier correction (m x 1), m = #constraints (here 2)

% --- Operators and dynamic residual (unconstrained form: M*ddq + H = tau)
[M, H, Kt, Ct, Cray] = model.operators(q, dq);
if isempty(Cray), Cray = 0 * Ct; end
Rdyn = M*ddq + H - pars.tau;

% --- Generalized-alpha effective tangent S_t = betap*M + gammap*(Ct+Cray) + Kt
St = pars.betap*M + pars.gammap*(Ct + Cray) + Kt;

```

```
% --- Constraint residual and linearization block B
phi = model.loop_phi(q);
Jc  = model.loop_Jc(q);       % size m x n  (m = 2 here)
if isfield(pars,'use_baumgarte') && pars.use_baumgarte
    % Acceleration-level stabilization:
    % rc = Jc*ddq + 2*zB*wB*(Jc*dq) + wB^2*phi
    rc = Jc*ddq + 2*pars.zB*pars.wB*(Jc*dq) + pars.wB^2*phi;
    % Corresponding linearization wrt q for the lower block:
    B  = pars.betap*Jc + pars.gammap*(2*pars.zB*pars.wB*Jc);
else
    % Pure position-level constraint in the Newton corrector
    rc = phi;
    B  = Jc;
end

% --- Mild regularization on the configuration block (improves robustness)
n = numel(q);
m = size(Jc,1);
if rcond(St) < 1e-12
    St = St + 1e-9*speye(n);
end

% --- Build and solve the coupled linear system:
%     [ St   Jc' ] [dq_step] = -[ Rdyn ]
%     [ B     0  ] [ dlam  ]   [  rc   ]
KKT  = [ St,  Jc.';
         B,   zeros(m,m) ];
rhs  = -[Rdyn; rc];

sol      = KKT \ rhs;
dq_step  = sol(1:n);
dlam     = sol(n+1:end);
end
```

Listing 5.5: One generalized-α Newton corrector with constraints (direct KKT solve).

5.13 Structural damping and selection of Rayleigh coefficients

A small amount of structural damping is applied only to the flexible degrees of freedom. In this way, spurious long-lasting vibrations of the beam are reduced, while the motion of the rigid links is left essentially unchanged. The classical

Rayleigh model is used on the flexible block,

$$\mathbf{C}_{\text{Ray}} \;=\; \alpha_R\,\mathbf{M}_{ff} \;+\; \beta_R\,\mathbf{K}_{ff}, \tag{5.56}$$

where $\mathbf{M}_{ff}$ and $\mathbf{K}_{ff}$ are, respectively, the mass and stiffness submatrices restricted to the flexible DOFs.
The two Rayleigh coefficients are chosen so that the first two bending modes of the flexible substructure have a prescribed damping ratio ζ (for instance $\zeta = 1\%$–2% in many engineering cases). The circular frequencies ω_1, ω_2 of these modes are obtained from the standard eigenproblem

$$\mathbf{K}_{ff}\,\phi = \omega^2\,\mathbf{M}_{ff}\,\phi \tag{5.57}$$

sorting the roots in increasing order. Matching ζ at ω_1 and ω_2 gives the well-known relations

$$\alpha_R \;=\; \frac{2\,\zeta\,\omega_1\omega_2}{\omega_1+\omega_2} \quad ; \quad \beta_R \;=\; \frac{2\,\zeta}{\omega_1+\omega_2} \tag{5.58}$$

where α_R has units of [s^{-1}] and β_R has units of [s]. For any other mode with circular frequency ω, the resulting damping ratio varies smoothly as

$$\zeta(\omega) \;=\; \tfrac{1}{2}\Big(\frac{\alpha_R}{\omega} + \beta_R\,\omega\Big).$$

This means that very low and very high modes remain only mildly damped, while the first two targeted modes are close to the desired ζ [13, 17]. Additional attenuation of very high frequencies comes from the generalized-α time integrator through the choice of ρ_∞ (see section 5.12).

5.14 Numerical conditioning and practical safeguards

Good numerical behaviour does not come for free. A few simple rules keep the solver stable and predictable:

5.14.1 Scaling

Degrees of freedom have different physical units (radians for the rigid angles, metres for the beam DOFs). A light scaling may be applied so that typical entries of the residuals and of the Jacobians have comparable magnitudes. This usually improves the behaviour of Newton iterations and of sparse factorizations [9, 10].

5.14.2 Constraint Jacobian

The loop Jacobian $\mathbf{J}_c(q)$ must keep full row rank (rank = 2) along the trajectory. When the mechanism approaches a nearly singular pose, reduce the time step and strengthen the position/velocity projection (increase the weights used to keep selected coordinates fixed). Monitoring $\mathrm{rank}(\mathbf{J}_c)$ and the norm of the constraint residual after projection is a simple and effective check.

5.14.3 Consistent linearizations

Quadratic convergence requires derivatives computed from the unsimplified expressions: form $\mathbf{K}_t = \partial\mathbf{H}/\partial q$ and $\mathbf{C}_t = \partial\mathbf{H}/\partial\dot{q}$ before algebraic cancellations. Early simplifications of $\mathbf{H}$ can remove terms that are essential in the derivatives and slow down, or even stall, the corrector.

5.14.4 Trust region on u, v

Limit the increment of axial and transverse beam DOFs within each Newton step (for instance, by a backtracking line search that caps the maximum change of u and v). This prevents unphysical jumps when the tangent is temporarily poor.

5.14.5 Mild regularization

If the iteration matrix $\mathbf{S}_t$ is close to singular (e.g. large pivot growth during a sparse LU (lower–upper) factorization of $\mathbf{S}_t$), add a small diagonal term proportional to the average diagonals of $\mathbf{M}$ and $\mathbf{K}_t$. This improves robustness without changing the solution in a noticeable way.

5.14.6 Stabilization coefficients

Choose the Baumgarte coefficients proportional to the time step (stabilization frequency $\omega_B = \kappa/\Delta t$ with a moderate κ) to avoid an unnecessarily stiff correction in the constraint equations [22].

5.15 Verification strategy

A short, focused verification plan helps isolate errors early and builds confidence in the model and in the code. The following checks are inexpensive and informative.

1. **Element-level patch tests.** (i) Axial extension of a single element: prescribe equal and opposite axial end forces and check that the displacement field is linear and that the internal force matches EA/L_e. (ii) Pure bending: prescribe

end moments to produce a constant curvature and check the rotation field and the bending energy. These tests validate the local matrices and the co-rotational mapping.

2. **Pinned–pinned modal test of the beam.** Isolate the flexible beam with pinned ends (rotation free, translations fixed) and solve $\mathbf{K}_{ff}\phi = \omega^2\mathbf{M}_{ff}\phi$. The first two circular frequencies ω_1, ω_2 should follow the expected trend with EI and ρA. Use them to set the Rayleigh coefficients via (5.58).

3. **Static lowering under gravity.** With gravity only, compute the static equilibrium of the isolated beam. The tip deflection should scale linearly with g and inversely with EI, and the deformed shape should remain smooth. This confirms signs and units in the gravitational load and the elastic energy.

4. **Closed-loop checks.** For the full mechanism, verify that $\text{rank}(\mathbf{J}_c) = 2$ at representative configurations and that the post-projection residual $\|\boldsymbol{\phi}(q)\|$ is small. Perform the congruence test in (5.37) to detect possible mistakes in local-to-global rotations, DOF ordering, or Boolean scatter.

5. **Tendon pretension path.** Vary the tendon rest length L_0 (or the prescribed shortening Δ) and solve the static problem with q_3 fixed (for instance $q_3 = 0$). The family of equilibria should evolve smoothly; a small, controlled imperfection on selected interior u-DOFs should select deterministically the bending side in the operating posture.

5.16 Simulations

In this section are available the dynamic simulations of the PACOMA leg model starting from the pre-tensioned state and the gravity force should lead to a free-fall behavior, as shown in Figure 5.11 where the simulation lasts 3 seconds. The MATLAB animation, shown in Figure 5.10, shows that the model cannot handle internal stresses well: the node deformations are enormous and the loop closure fails.
Complete videos are available in the GitLab repository cited at the end of chapter 1.

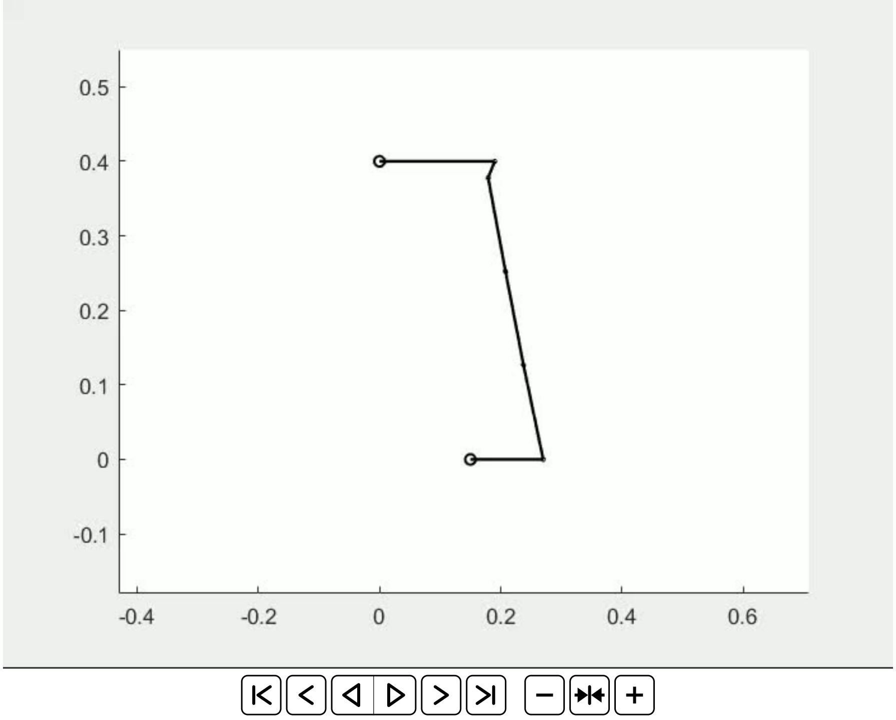


Figure 5.10: Animation of MATLAB simulation (free-fall after pretension).

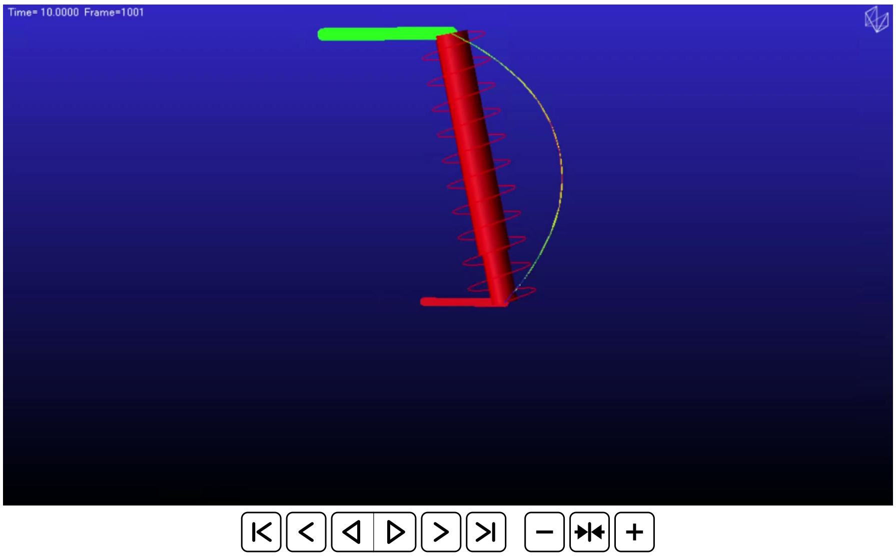


Figure 5.11: Animation of ADAMS simulation (free-fall after pretension).

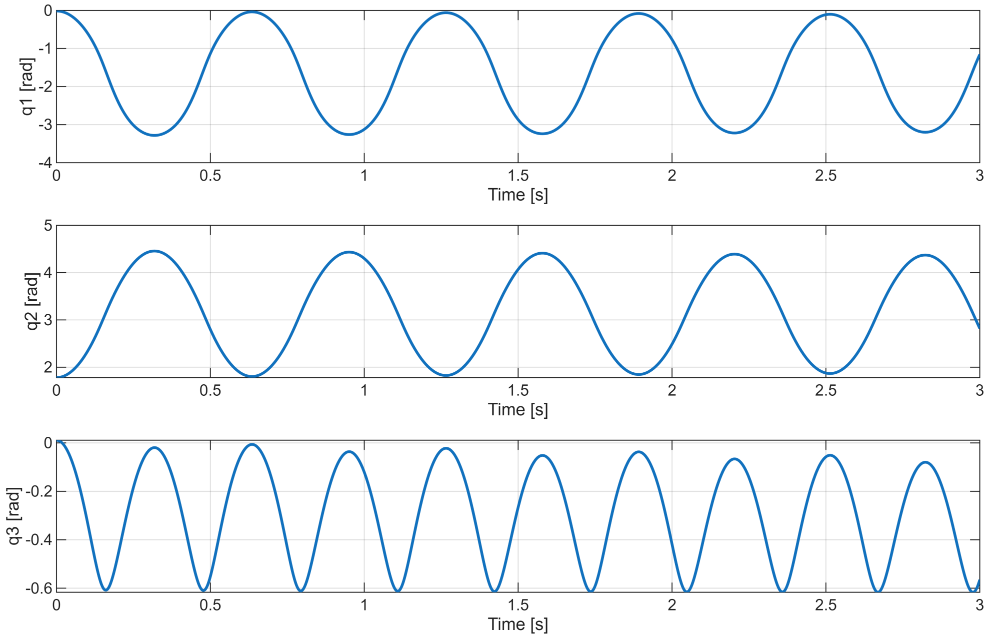


Figure 5.12: Angles q_1, q_2 and q_3 during the 3 seconds free-fall simulation after the pretensioning transient.

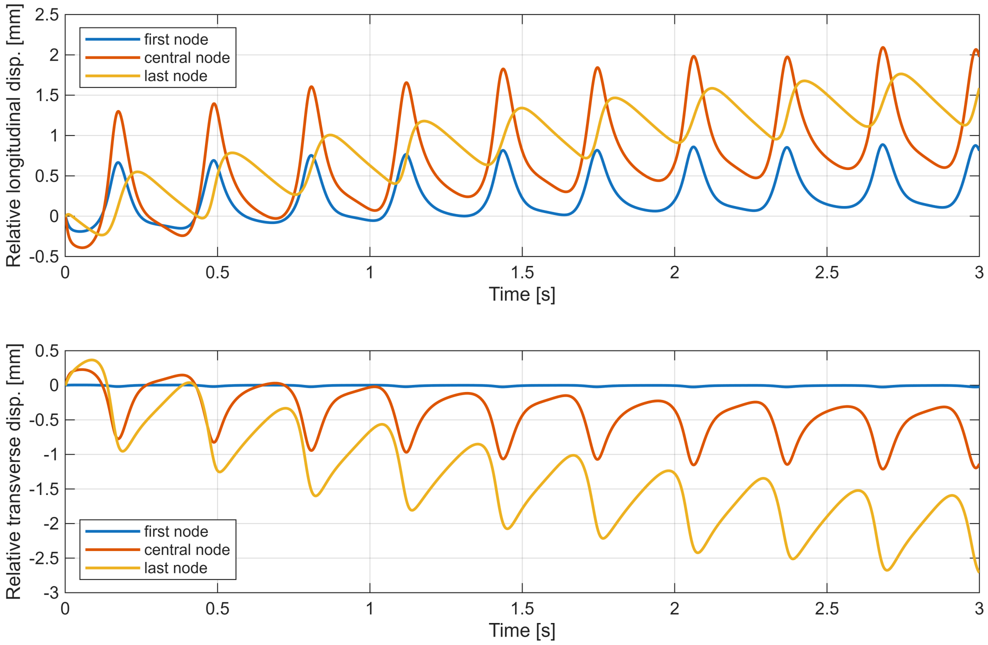


Figure 5.13: The displacement of the three internal nodes of the flexible beam (when it is divided into 4 element) during the 3 seconds free-fall simulation after the pretensioning transient.

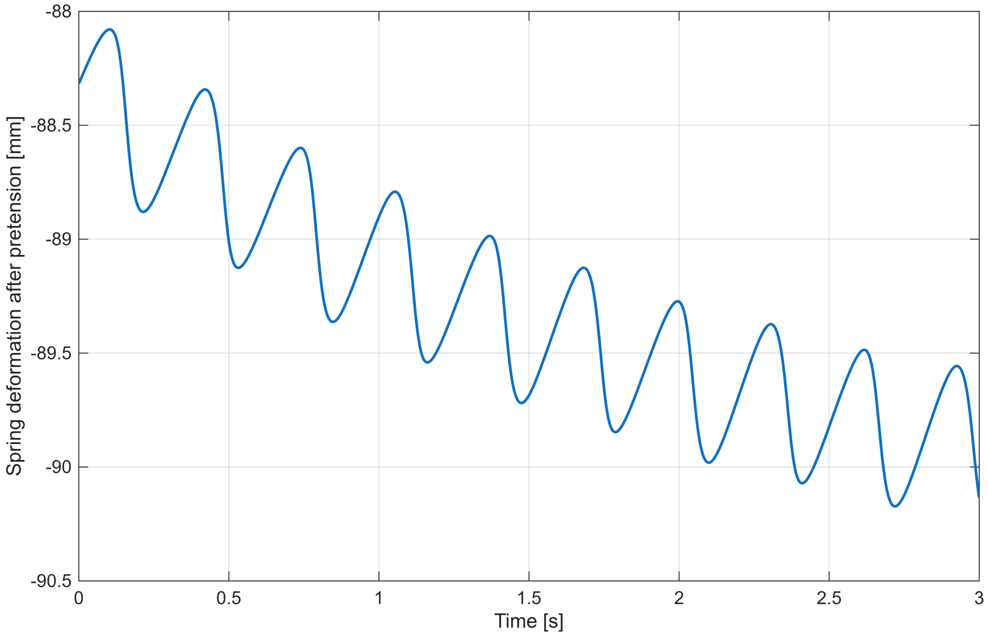


Figure 5.14: Spring deformation during the 3 seconds free-fall simulation after the pretensioning transient.

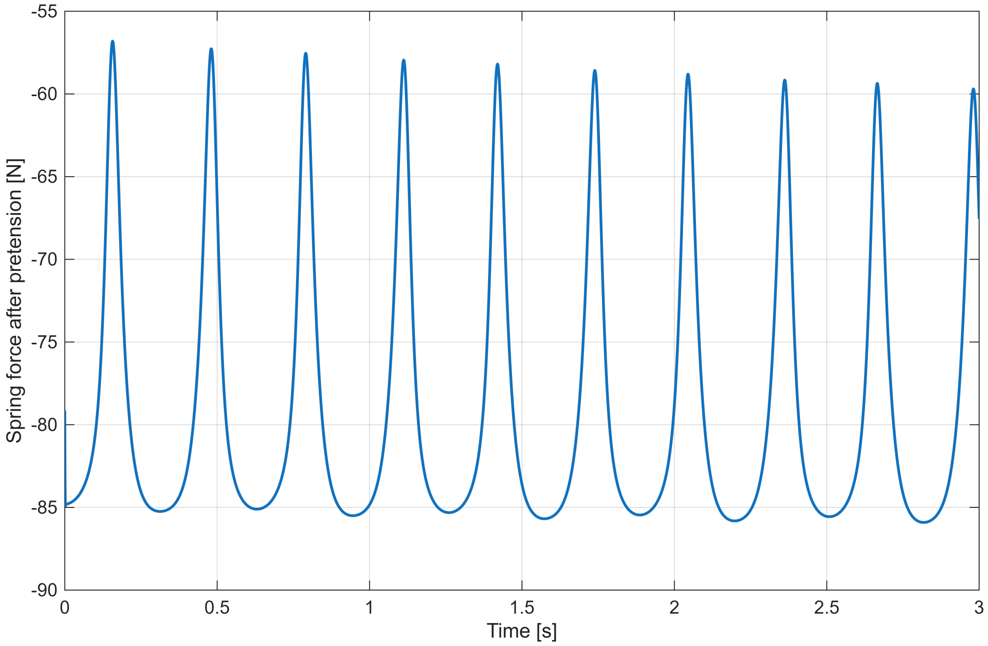


Figure 5.15: Spring force during the 3 seconds free-fall simulation after the pretensioning transient.

Chapter 6

Conclusions and Future Work

6.1 Summary of contributions

This work has presented a planar rigid–flexible leg model composed of two rigid links connected by revolute joints and a flexible Timoshenko beam modeled by a co–rotational finite element discretization. The model includes gravity, elastic strain energy with a von Kármán enrichment for moderate rotations, and a tendon that links the lower rigid link to the beam tip through a linear spring–dashpot. The loop closure between the beam tip and the upper link is enforced by Lagrange multipliers, with optional acceleration–level stabilization and projection of positions and velocities to keep the constraints satisfied. The equations of motion are integrated in time by the generalized–α scheme, with consistent linearizations and an optional Rayleigh damping applied to the flexible degrees of freedom only. A verification plan has been outlined to check element matrices, co–rotational mapping, modal properties, gravity sag, loop rank, and tendon pretension.

The main outcome is a coherent modeling pathway that starts from kinematics, proceeds through energy terms, delivers a constrained set of equations suitable for implicit time stepping, and is supported by practical safeguards for numerical robustness. While the present implementation focuses on a single planar leg, the structure of the formulation lends itself to extensions in several directions, discussed next.

6.2 Limitations and possible extensions

The current formulation has several deliberate simplifications.

Planar setting

Only planar motion is modeled. In situations where out–of–plane effects, torsion of the flexible member, or three–dimensional tendon routing are relevant, a spatial extension is required. A practical path is to retain the same co–rotational idea in 3D (element frames attached to current chords) and to promote the scalar rotation θ to a full spatial rotation (e.g. Tait–Bryan angles or quaternions) with an appropriate update rule and consistent mass and stiffness terms [16, 13].

Timoshenko with von Kármán enrichment

The beam model assumes small strains in the element local frames, with geometric nonlinearity handled by a von Kármán axial term. For very large rotations with significant curvature redistribution, geometrically exact beam formulations of the Simo–Reissner type offer a more faithful description at comparable mesh densities [29]. For very slender members where axial stretching and curvature coupling are dominant or where path–following through strong geometric softening is required, absolute nodal coordinate formulations (ANCF) are an option, at the expense of more degrees of freedom and stronger nonlinearity, significantly increasing the computational effort [21].

Actuation and friction

The tendon is modeled as an ideal spring–dashpot acting along its current direction. In real systems, friction at guides and pulleys, slack–take–up, and saturation of the actuation appear. These effects can be added by replacing the linear spring with a more detailed cable model (piecewise straight segments with friction coefficients) and by including actuator limits and rate limits in the generalized force. Contact at mechanical hard stops or at interfaces can be added with gap functions and complementarity constraints.

Computational cost and implementation efficiency

The present implementation is primarily designed for clarity and consistency rather than for speed. In particular, the co–rotational description requires updating the orientation of every beam element, rebuilding local frames, and recomputing the corresponding local–to–global transformations and constraint Jacobians at each time step and at every Newton iteration. Together with the use of consistent linearizations of all terms in the generalized–α scheme, this makes each solver step relatively expensive and results in long runtimes for simulations of practical duration.

At this stage the codes provide a faithful reference implementation, but they are computationally heavy and not yet suited for systematic parameter sweeps or real–time applications. A natural line of future work is therefore to focus on efficiency: profiling the current routines, reorganizing data structures, vectorizing operations over elements, exploiting code generation or compiled back–ends, and possibly introducing reduced–order models for the flexible dynamics. These steps would preserve the structure of the formulation while significantly speeding up the time integration.

6.3 LQR control in a nutshell

This section outlines a simple path to add a stabilizing state–feedback without diving into implementation details. The idea is to regulate small motions around a chosen operating posture by linearizing the model and applying a linear–quadratic regulator (LQR) [30, 31].

6.3.1 1) Pick an operating point.

Select a static posture $q^\star$ (for instance the pretensioned shape with $q_3 = 0$) that satisfies the loop closure and static balance. The controller will act only for small deviations around this posture.

6.3.2 2) Obtain a small linear model.

Linearize the constrained dynamics around $(q^\star, \dot{q}^\star = \mathbf{0})$ and keep a small set of independent coordinates x (for example, the platform angle and one or two flexible modal coordinates). In compact form:

$$\dot{z} = \mathbf{A}\, z + \mathbf{B}\, u \quad , \quad z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{6.1}$$

where u is the control input (e.g. a tendon force or a motor torque). The linearization can be produced either by eliminating the constraints with a projection basis, or by forming and reducing the linearized KKT system; both routes give an equivalent $(\mathbf{A}, \mathbf{B})$ near $q^\star$.

6.3.3 3) Choose the LQR weights.

Select two symmetric weighting matrices $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$. The matrix $\mathbf{Q}$ tells the controller which states should stay small (typically the rigid angle to be held and the first flexible mode), and $\mathbf{R}$ penalizes aggressive actuation. A practical start is to put moderate positive values on the regulated positions and velocities in $\mathbf{Q}$, and to set $\mathbf{R}$ as a positive scalar that reflects the actuator limits.

6.3.4 4) Solve the Riccati equation and form the gain.

Solve the continuous algebraic Riccati equation (CARE)

$$\mathbf{A}^\top\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{6.2}$$

and set the feedback $u = -\mathbf{K}z$ with $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}$. If the controller runs at a fixed sampling time, use the discrete-time CARE instead [30].

6.3.5 5) Keep it gentle with flexible modes.

To avoid exciting neglected high–frequency dynamics (spillover), keep the closed–loop bandwidth comfortably below the first flexible mode not included in z, and penalize modal velocities in $\mathbf{Q}$ to add damping. If needed, add a light structural damping in the plant model (Rayleigh on the flexible subspace) and retune. These rules of thumb are standard when controlling flexible structures [32, 33].

6.3.6 6) Test and iterate.

Implement $u = -\mathbf{K}z$ in the nonlinear simulator, start with very small perturbations, and increase the test envelope gradually. If not all components of z are measured, add a simple observer (e.g. Luenberger or Kalman) using encoders and a tip estimate; keep the observer bandwidth below the first neglected flexible mode.

This procedure gives a controller that holds the leg in its operating posture, damps the dominant flexible motion, and respects actuator effort through $\mathbf{R}$. More advanced designs (multi–objective, robust, or centralized across several legs) can be built on the same linearized blocks if needed.

6.4 From one leg to a six–leg parallel arrangement

The final goal in PACOMA is a parallel continuum manipulator with multiple flexible legs anchoring a payload or a docking ring [1]. Moving from a single planar leg to a six–leg arrangement (in 2D or, more realistically, in 3D) can follow a modular path.

Model structure

Replicate the present leg model N_ℓ times (e.g. $N_\ell = 6$). Each leg i has its own state $q^{(i)}$ and its own tendon model. The legs connect a rigid base to a rigid platform. Add the platform pose p (position and, in 3D, orientation) to the generalized coordinates, and write one loop–closure for each leg that attaches its beam tip to the corresponding platform anchor. The full constraint stack has size equal to twice the number of legs in 2D (or three times in 3D). The mass matrix is block diagonal in the leg blocks plus the platform block; the coupling between legs enters entirely through the platform constraints. The global KKT system is

$$\begin{bmatrix} \mathrm{blkdiag}(\mathbf{S}_t^{(1)}, \dots, \mathbf{S}_t^{(N_\ell)}, \mathbf{S}_t^{\mathrm{plat}}) & \mathbf{J}_c^\top \\ \mathbf{J}_c & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta q^{\mathrm{all}} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{R}_{\mathrm{dyn}}^{\mathrm{all}} \\ -\mathbf{r}_c \end{bmatrix},$$

with a sparse structure well suited for Schur–complement solvers. As the number of legs grows, re–ordering and preconditioning become important.

Platform kinematics and geometry

Choose base and platform anchor locations. In 3D, use standard rigid–body parameterizations for orientation (unit quaternions or exponential coordinates). The Jacobians of the leg–to–platform constraints are obtained by differentiating the platform anchor maps with respect to platform pose, and by reusing the leg tip Jacobians already present in the single–leg model.

Actuation and allocation

If several tendons (one per leg) act on the platform, there is redundancy. At the control level, map desired platform wrenches into tendon forces by solving a small allocation problem that minimizes, for example, the sum of squared tendon forces subject to lower/upper bounds and positivity of tension. In the model, represent tendon forces as generalized forces on the leg and on the platform with equal magnitude and opposite direction along each cable.

Control architectures

There are two main options.

- *Centralized.* Linearize the full reduced model of platform plus legs, and design a single LQR (or $\mathcal{H}_\infty$) that stabilizes the platform pose while damping the first flexible modes of all legs. This gives the best performance but increases the order of the design model.

- *Decentralized with coordination.* Design one low–order LQR for the platform (using a rigidified model) and several local LQRs per leg to damp flexible motion. Then superimpose the platform command and the leg damping commands. This is simpler and scales well. Care is needed to avoid fighting actions; a simple priority scheme or a null–space projector for the leg damping forces works in practice [32].

From plane to space

A realistic six–leg PACOMA is spatial. The extension follows the same ideas: geometrically exact or 3D co–rotational beams for the legs, a 6–DoF platform, and three scalar constraints per leg. The tendon routing must be modeled in space, including friction at guides if present. Standard references on parallel manipulators can guide anchor placement and conditioning [34].

Computing effort

The cost per time step grows with the number of elastic coordinates and number of legs. Even so, the KKT matrix is sparse and has a repeating block structure. A Schur complement on the small multiplier block, combined with sparse LU or iterative solvers, keeps the cost manageable. For real–time or hardware–in–the–loop tests, code generation, vectorization, and parallel assembly are recommended.

6.5 Concluding remarks

The present planar rigid–flexible leg model provides a solid basis to understand the interplay between rigid–body motion, flexible deformation, tendon forces, and loop closure in a PACOMA–like mechanism. The modeling choices are conservative and intentionally simple, yet they capture the main physical effects and allow a direct path toward control. The next steps are clear: (i) close the loop with an LQR designed on the reduced linear model near the operating posture; (ii) validate the predictions by comparison with an independent multibody tool (e.g. ADAMS) and, where possible, with experiments on a laboratory prototype; (iii) scale the formulation to multiple legs and to three dimensions. These developments, carried out with the same attention to consistent linearizations, constraint handling, and numerical safeguards, will enable a full–order model of a parallel continuum manipulator and pave the way to guidance and control for docking tasks in space [1].

Bibliography

[1] Bingbin Yu, Yu-Hung Pai, Marc Manz, Martin Garland, Jonathan Babel, Petri T. Piiroinen, Frank Kirchner, and Shivesh Kumar. *PACOMA: Parallel Continuum Manipulator for Docking in Space*. Manuscript / project report. DFKI, Chalmers University of Technology, Airbus D&S. 2024 (cit. on pp. 1–3, 5, 45, 57, 64, 85, 106, 108).

[2] V. Rodrigues, B. Yu, C. Stoeffler, and S. Kumar. «Kinetostatic Analysis for 6RUS Parallel Continuum Robot Using Cosserat Rod Theory». In: *Advances in Robot Kinematics 2024 (ARK 2024)*. Ed. by J. Lenarčič and M. Husty. Vol. 31. Springer Proceedings in Advanced Robotics. Cham, Switzerland: Springer, 2024. DOI: `10.1007/978-3-031-64057-5_49` (cit. on pp. 1, 2, 5, 57).

[3] *MATLAB (R2025b)*. The MathWorks, Inc. Natick, MA, USA, 2025. URL: `https://www.mathworks.com` (visited on 11/10/2025) (cit. on p. 5).

[4] *Adams View User's Guide (Adams 2022)*. MSC Software Corporation. Newport Beach, CA, USA, 2022. URL: `https://www.mscsoftware.com` (cit. on p. 5).

[5] S. Timoshenko. «On the correction for shear of the differential equation for transverse vibrations of prismatic bars». In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 41 (1921), pp. 744–746 (cit. on pp. 5, 21, 66, 74, 75).

[6] J. Chung and G. M. Hulbert. «A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized-α Method». In: *Journal of Applied Mechanics* 60.2 (1993), pp. 371–375. DOI: `10.1115/1.2900803` (cit. on pp. 6, 15, 21, 31, 93–95).

[7] S. Kumar, K. A. von Szadkowski, A. Müller, and F. Kirchner. «An Analytical and Modular Software Workbench for Solving Kinematics and Dynamics of Series-Parallel Hybrid Robots». In: *Journal of Mechanisms and Robotics* 12.2 (2020), p. 021114. DOI: `10.1115/1.4045941` (cit. on p. 8).

[8] Rémi Servajean. *Simulation of a Parallel Continuum Manipulator for Space Docking*. Summer Internship Report. Supervisors: Shivesh Kumar, François Dufour. Chalmers University of Technology and Bordeaux INP, Aug. 2024 (cit. on p. 8).

[9] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential and Differential-Algebraic Equations*. SIAM, 1998 (cit. on pp. 14, 17, 68, 82, 83, 95, 97).

[10] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 2nd. Springer, 1996 (cit. on pp. 14, 17, 68, 82, 83, 95, 97).

[11] R. D. Mindlin. «Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates». In: *Journal of Applied Mechanics* 18 (1951), pp. 31–38 (cit. on p. 21).

[12] G. R. Cowper. «The Shear Coefficient in Timoshenko's Beam Theory». In: *Journal of Applied Mechanics* 33.2 (1966), pp. 335–340. DOI: `10.1115/1.3625046` (cit. on pp. 21, 25, 66).

[13] K. J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996 (cit. on pp. 21, 25, 29, 30, 66, 74–76, 78, 93–95, 97, 104).

[14] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover, 2012 (cit. on pp. 21, 66, 74–76, 78).

[15] J. N. Reddy. *An Introduction to the Finite Element Method*. 3rd. McGraw-Hill, 2004 (cit. on pp. 21, 25, 29, 73, 76, 78, 79, 92).

[16] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. 2nd. Wiley, 2013 (cit. on pp. 21, 29, 30, 66, 73, 76, 79, 92, 104).

[17] M. Geradin and D. Rixen. *Mechanical Vibrations: Theory and Application to Structural Dynamics*. 3rd. Wiley, 2015 (cit. on pp. 21, 30, 67, 80, 93, 94, 97).

[18] N. M. Newmark. «A Method of Computation for Structural Dynamics». In: *Journal of the Engineering Mechanics Division* 85.3 (1959), pp. 67–94 (cit. on pp. 21, 31).

[19] *Validation Case: Modal Analysis of a Cantilever Beam*. OnScale Solve Help. OnScale. 2023. URL: `https://onscale.com/help/solve/validation/validation-case-modal-analysis-cantilever-beam/` (visited on 11/04/2025) (cit. on p. 32).

[20] M. A. Crisfield. «A Consistent Co-Rotational Formulation for Non-Linear, Three-Dimensional, Beam-Elements». In: *Computer Methods in Applied Mechanics and Engineering* 81.2 (1990), pp. 131–150 (cit. on p. 66).

[21] Ahmed A. Shabana. *Computational Dynamics*. 3rd. Wiley, 2013 (cit. on pp. 67, 68, 76, 80, 82, 104).

[22] O. A. Bauchau. *Flexible Multibody Dynamics*. Springer, 2011 (cit. on pp. 68, 82, 98).

[23] Theodore von Kármán. «Festigkeitsprobleme im Maschinenbau». In: *Encyklopädie der Mathematischen Wissenschaften* IV.4 (1910), pp. 311–385 (cit. on p. 73).

[24] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 4th. Johns Hopkins University Press, 2013 (cit. on p. 80).

[25] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. 2nd. Cambridge University Press, 2013 (cit. on p. 80).

[26] J. Baumgarte. «Stabilization of Constraints and Integrals of Motion in Dynamical Systems». In: *Computer Methods in Applied Mechanics and Engineering* 1.1 (1972), pp. 1–16 (cit. on p. 82).

[27] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. 1st. London: Springer, 2009. ISBN: 978-1-84996-634-8 (cit. on p. 86).

[28] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2006. ISBN: 978-0-471-64990-8 (cit. on p. 86).

[29] J. C. Simo and L. Vu-Quoc. «A Three-Dimensional Finite-Strain Rod Model. Part II: Computational Aspects». In: *Computer Methods in Applied Mechanics and Engineering* 58.1 (1986), pp. 79–116 (cit. on p. 104).

[30] Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. SIAM, 2007 (cit. on pp. 105, 106).

[31] Frank L. Lewis, Draguna Vrabie, and Vassilis L. Syrmos. *Optimal Control*. 3rd. Wiley, 2012 (cit. on p. 105).

[32] André Preumont. *Vibration Control of Active Structures*. 4th. Springer, 2018 (cit. on pp. 106, 108).

[33] Mark J. Balas. «Feedback control of flexible systems». In: *IEEE Transactions on Automatic Control* 23.4 (1978), pp. 673–679. DOI: `10.1109/TAC.1978.1101814` (cit. on p. 106).

[34] Jean-Pierre Merlet. *Parallel Robots*. 2nd. Springer, 2006 (cit. on p. 108).