

# SUPPLEMENTARY MATERIAL of paper: A Retractile Tendon-Sensor Continuum Robot with Embedded Proprioception

Allan Coronado \* Hugo A. Moreno \* Rodolfo Verdín \*  
 Juan E. Velázquez-Velázquez \*\* Gerardo Flores \*\*\*

\* Laboratorio de Percepción y Robótica, Center for Research in Optics,  
 León 37150, Mexico, (e-mail:  
 allanca@cio.mx, hugoamj@cio.mx, rverdin@cio.mx)

\*\* Unidad Profesional Interdisciplinaria de Ingeniería Campus  
 Hidalgo, Instituto Politécnico Nacional, San Agustín Tlaxiaca,  
 Hidalgo, 42162, Mexico, (e-mail: jvelazquezv@ipn.mx)

\*\*\* RAPTOR Lab, School of Engineering, College of Arts and  
 Sciences, Texas A&M International University, Laredo, TX 78041  
 USA, (e-mail: gerardo.flores@tamiu.edu)

## Simulator Material

The simulation code, 3D models of the continuum and retractile soft arm, and demonstration files—including Python and MATLAB implementations for forward/inverse kinematics and Jacobian-based control with dynamic extension—are provided at: [GitHub Continuum-and-retractive-SOFT-ROBOT](#)

### 1. GRIPPER



Fig. 1. TPU flexible gripper integrated at the distal end of the retractile tendon–sensor continuum arm. The gripper is actuated by a DC motor (105 RPM) that winds three tendons, one per finger, producing a closing motion for object grasping. When the motor unwinds, the elastic TPU structure restores the fingers to their initial open position.

A soft gripper was designed and fabricated using thermoplastic polyurethane (TPU) through 3D printing. The gripper is mounted at the distal end of the continuum manipulator and actuated by a miniature DC motor rated at 105 RPM—the same model employed for the body tendons. This motor winds three independent tendons, one attached to each finger of the gripper. As the motor rotates, the tendons are simultaneously tensioned, causing the fingers to bend and close around the object. When the

motor reverses, the tendons are released, and the intrinsic elasticity of the TPU material restores the fingers to their open configuration. This mechanism enables adaptive, compliant grasping while maintaining full mechanical and electrical compatibility with the continuum arm’s tendon-driven architecture.

### 2. CUSTOM ELECTRONIC INTERFACE BOARD

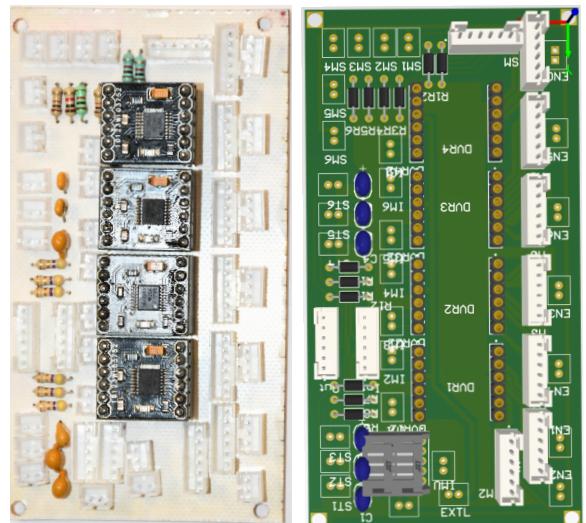


Fig. 2. Custom electronic interface board for the retractile tendon–sensor continuum robot. The board integrates motor drivers, sensor conditioning circuits, and regulated power distribution into a single platform. This reduces wiring complexity, ensures consistent grounding across channels, and improves overall robustness during experiments.

A custom electronic interface board was developed to consolidate the control and sensing electronics of the manipulator into a single, compact module. The board routes all input and output channels from the microcontroller, distributes regulated power to the six motor drivers, and

supplies excitation signals for the nichrome–nylon tendon sensors. It also includes analog conditioning stages for tendon resistance, tendon-tension transducers, and inertial measurement units. This integration minimizes wiring between the arm and controller, reduces electromagnetic interference, and ensures a stable common ground reference for all sensing paths.

### 3. TENDON-SENSOR CALIBRATION: RESISTANCE-LENGTH RELATIONSHIP

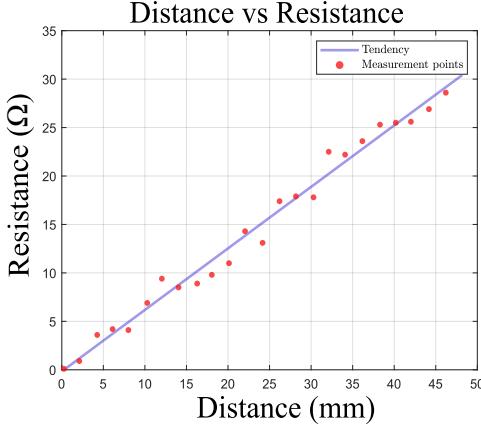


Fig. 3. Calibration of the nichrome–nylon tendon sensor. The resistance increases linearly with physical length, allowing real-time estimation of tendon elongation. The linear fit (black line) achieved an RMSE of 1.19 mm within the 35–71 mm operating range.

Bench calibration was performed to characterize the linear relationship between the tendon’s physical length and its electrical resistance. Known elongations were applied to the nichrome–nylon braid while recording resistance values, yielding the calibration curve shown in Fig. 3. Within the operating range  $L \in [35, 71]$  mm, the filament behaves as a linear transducer described by

$$R(L) = R_0 + k_L(L - L_0), \quad k_L > 0, \quad (1)$$

with an RMSE of 1.19 mm. This calibration was later used to map ADC readings to tendon length and to verify linearity and repeatability under pulsed excitation.

### 4. SENSOR FUSION FORMULATION

The complete proprioceptive estimate  $\hat{\mathbf{x}}$  is obtained through a weighted, constrained least-squares problem that fuses tendon length, tension, encoder, and inertial measurements:

$$\begin{aligned} \hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{X}} & \left[ \sum_{i=1}^6 w_L \|L_i - L_i(\mathbf{x})\|^2 + \sum_{i=1}^6 w_F \psi(F_i) \right. \\ & \left. + \sum_{i=1}^6 w_E \|L_{\text{enc},i} - L_i(\mathbf{x})\|^2 + w_I \varphi(\mathbf{q}_b, \mathbf{q}_t; \mathbf{x}) \right], \end{aligned} \quad (2)$$

where  $\mathcal{X}$  enforces tendon limits  $L \in [35, 71]$  mm and the weights  $(w_L, w_F, w_E, w_I)$  define sensor confidence. The functions  $\psi$  and  $\varphi$  penalize tension-limit violations and inertial inconsistencies, respectively. The resulting  $\hat{\mathbf{x}}$  and its derivative are used for control, while the tension envelope and integrity flags enforce immediate safety stops.

### 5. STABILITY FOR CONSTANT TARGETS

Assume  $\dot{\mathbf{P}}_d \equiv \mathbf{0}$  and that  $\mathbf{J}(\mathbf{L})$  has full row rank in the operating region. Then  $0 < \lambda_{\min}(\mathbf{\Pi}_\lambda) =: \underline{\pi} \leq \bar{\pi} := \lambda_{\max}(\mathbf{\Pi}_\lambda) < 1$ . Consider the Lyapunov function

$$V(\mathbf{e}, \mathbf{r}) = \frac{1}{2}\|\mathbf{e}\|^2 + \frac{1}{2}\|\mathbf{r}\|^2. \quad (3)$$

Its derivative along (22)–(24) is

$$\dot{V} = \mathbf{e}^\top \dot{\mathbf{e}} + \mathbf{r}^\top \dot{\mathbf{r}} = -\mathbf{e}^\top \mathbf{\Pi}_\lambda \Lambda \mathbf{e} - \mathbf{e}^\top \mathbf{\Pi}_\lambda \mathbf{r} - \mathbf{r}^\top K_d \mathbf{r} + \mathbf{r}^\top \Lambda \mathbf{e}. \quad (4)$$

Using  $\|\mathbf{\Pi}_\lambda\| = \bar{\pi}$ ,  $\lambda_{\min}(\Lambda) =: \underline{\lambda}$ ,  $\lambda_{\min}(K_d) =: \underline{k}$ , and the Cauchy–Schwarz/Young inequality  $2a^\top b \leq \rho\|a\|^2 + \rho^{-1}\|b\|^2$  (any  $\rho > 0$ ), we obtain

$$\begin{aligned} \dot{V} &\leq -\underline{\pi} \underline{\lambda} \|\mathbf{e}\|^2 - \underline{k} \|\mathbf{r}\|^2 + \bar{\pi} \|\mathbf{e}\| \|\mathbf{r}\| + \|\Lambda\| \|\mathbf{e}\| \|\mathbf{r}\| \\ &\leq -\underline{\pi} \underline{\lambda} \|\mathbf{e}\|^2 - \underline{k} \|\mathbf{r}\|^2 + c \|\mathbf{e}\| \|\mathbf{r}\| \quad (c := \bar{\pi} + \|\Lambda\|) \\ &\leq -\left(\underline{\pi} \underline{\lambda} - \frac{\rho}{2}\right) \|\mathbf{e}\|^2 - \left(\underline{k} - \frac{c^2}{2\rho}\right) \|\mathbf{r}\|^2. \end{aligned} \quad (5)$$

Choose  $\rho \in (0, 2\underline{\pi} \underline{\lambda})$  and gains so that  $\underline{k} > \frac{c^2}{2\rho}$ . Then there exist  $\alpha_1, \alpha_2 > 0$  such that

$$\dot{V} \leq -\alpha_1 \|\mathbf{e}\|^2 - \alpha_2 \|\mathbf{r}\|^2 \leq -\alpha V, \quad \alpha := 2 \min\{\alpha_1, \alpha_2\},$$

which proves **local exponential stability** of  $(\mathbf{e}, \mathbf{r}) = (\mathbf{0}, \mathbf{0})$ . Hence  $\mathbf{e}(t) \rightarrow \mathbf{0}$  exponentially.

*Tracking of time-varying targets*

For general  $\dot{\mathbf{P}}_d \neq \mathbf{0}$  in (24), the same Lyapunov function gives

$$\dot{V} \leq -\alpha_1 \|\mathbf{e}\|^2 - \alpha_2 \|\mathbf{r}\|^2 + \|\mathbf{e}\| \|\mathbf{d}(t)\|.$$

If  $\dot{\mathbf{P}}_d$  is bounded, then  $\mathbf{d}(t)$  is bounded (and vanishes as  $\lambda \rightarrow 0$  and away from singularities, where  $\mathbf{\Pi}_\lambda \rightarrow \mathbf{I}_3$ ).

<sup>1</sup> Therefore the origin is *input-to-state stable* (ISS); in particular, one obtains

$$\|(\mathbf{e}(t), \mathbf{r}(t))\| \leq \kappa e^{-\alpha t} \|(\mathbf{e}(0), \mathbf{r}(0))\| + \gamma \left( \sup_{\tau \leq t} \|\mathbf{d}(\tau)\| \right),$$

for class- $\mathcal{K}$  function  $\gamma$  and constants  $\kappa, \alpha > 0$  determined by the gains. If  $\dot{\mathbf{P}}_d$  is known and  $\lambda$  is small (or  $\lambda = 0$  with full row rank so that  $\mathbf{\Pi}_\lambda = \mathbf{I}_3$ ), then  $\mathbf{d} \equiv \mathbf{0}$  and the tracking error is again **locally exponentially stable**.

**Remark 1.** The filter (22) plays the role of an integral action in kinematics; the pair  $(\Lambda, K_d)$  tunes proportional/damping levels. This is why (23) behaves as a PI controller in task space.

No null-space motion is injected. Under DLS it would appear in the error channel and degrade the bound above; if a secondary task is required, keep it bounded and its gain small to preserve practical stability.

**Remark 2.** If  $\mathbf{P}_d(t)$  is time-varying and  $\dot{\mathbf{P}}_d$  is unknown, then a bounded disturbance term appears in  $\dot{V}$ , leading to input-to-state stability: the error remains bounded and converges to a neighborhood of zero, with radius proportional to the supremum of  $\|\dot{\mathbf{P}}_d\|$ .

### 6. SIMULATION

All simulations were carried out in the MATLAB R2024a environment using a custom numerical script. The model

<sup>1</sup> Here  $\lambda$  is fixed and positive. The notation “ $\lambda \rightarrow 0$ ” is only used in a limiting sense:  $\mathbf{\Pi}_\lambda \rightarrow \mathbf{J}\mathbf{J}^\dagger$  as  $\lambda \rightarrow 0$ , recovering the exact projector onto  $\text{Im}(\mathbf{J})$ , and thus  $(\mathbf{I} - \mathbf{\Pi}_\lambda)\dot{\mathbf{P}}_d$  vanishes away from singularities.

represents the kinematic and control behavior of a two-section soft robotic arm actuated by six tendons. The simulator is available at: [GitHub Continuum-and-retractive-SOFT-ROBOT](#).

The simulation was performed with a sampling time of  $dt = 0.01$  s, corresponding to a total simulation time of  $T_{\text{sim}} = 17$  s. The Euler explicit integration method was used to update the tendon lengths  $L_i$ , ensuring real-time computational efficiency. The solver operates in discrete-time form, and no additional Simulink blocks were used.

The initial tendon configuration of the manipulator was defined as:

$$L_0 = [60 \ 60 \ 60 \ 60 \ 35 \ 50]^T \text{ mm.}$$

At each time step, the forward kinematics were computed to obtain the current end-effector position, and the control law was applied in Cartesian space.

A circular trajectory was selected as the desired end-effector path, parameterized as:

$$\mathbf{P}_d = \begin{bmatrix} X_d + R \cos(\omega t) \\ Y_d + R \sin(\omega t) \\ Z_d \end{bmatrix},$$

with radius  $R = 40$  mm, angular velocity  $\omega = 3\pi/20$  rad/s, and  $(X_d, Y_d, Z_d) = (-32.35, 37.63, 75.20)$ .

The proportional and derivative gains were set to  $K_p = 50$ ,  $K_d = 0.03$ , and  $\lambda = 0.1$ , respectively. A velocity saturation was applied to each tendon with a maximum rate of change of  $\dot{L}_{\max} = 8$  mm/s. Numerical derivatives were computed using finite differences, and the Jacobian was updated at each iteration using a small perturbation step  $\varepsilon = 10^{-3}$ .

The simulation results include the time evolution of tendon lengths and end-effector trajectories, as well as a 3D visualization of the manipulator's final configuration.

### 6.1 Simulation Results

Fig. 4 summarizes the simulation outcomes obtained under the Jacobian-based control with dynamic extension. Subfigure (a) illustrates the soft manipulator's final 3D configuration after completing the circular trajectory, highlighting the spatial deformation achieved through coordinated tendon actuation. The solid colored traces represent the backbone and segment positions, while the dashed and bold lines denote the desired and actual end-effector trajectories, respectively. The three faint, thin curves visible on the sides of the panel are the orthogonal projections of the desired and actual trajectories onto the  $XY$ ,  $XZ$ , and  $YZ$  planes, included to provide depth perception.

Subfigure (b) presents the corresponding time-domain evolution of the end-effector position, Cartesian tracking errors, and tendon length variations. A small residual oscillation is observed in the error signals due to numerical differentiation and discretization effects in the Jacobian update.

As observed in the top plot of Fig. 4b, the end-effector positions  $X_f$ ,  $Y_f$ , and  $Z_f$  closely follow their sinusoidal references, describing the desired circular motion in Cartesian space. Each oscillation period of approximately 8 s corresponds to one full revolution, yielding two complete

cycles within the total simulation time of 17 s. The middle plot shows that the Cartesian tracking errors converge rapidly to near-zero values after an initial transient of about 2 s, demonstrating the controller's capability to effectively compensate for nonlinear coupling while maintaining smooth and accurate trajectory tracking. Finally, the bottom plot depicts the coordinated tendon length variations required to generate the 3D circular motion, all remaining within feasible physical limits to ensure continuous deformation without excessive tendon strain.

## 7. VALIDATION

The proposed dynamic model and control strategy were experimentally validated by comparing the predicted and measured positions and orientations of the robotic arm. For this purpose, a dedicated stereo vision-based test bench was developed to capture the three-dimensional posture of each manipulator link during controlled motion.

The system consists of two calibrated and synchronized stereo cameras, positioned frontally and laterally to obtain redundant and complementary information. The stereo image processing provides the spatial coordinates  $(x, y, z)$  and the orientations in the principal planes, enabling an accurate 3D reconstruction of each link.

Fig. 5 presents the general layout of the test bench. The front stereo camera is responsible for determining the position  $(x, y, z)$  and the orientation in the  $xz$  plane of the centroid corresponding to the terminal section of each link, with respect to the robot's reference frame. Complementarily, the side stereo camera estimates the position  $(x, y, z)$  and the orientation in the  $yz$  plane. This redundancy in centroid measurement helps mitigate errors caused by potential occlusions or interlink interference, by automatically selecting the most reliable measurement and discarding those affected by visibility loss.

Finally, the complete three-dimensional orientation of each link is obtained by fusing the partial orientations estimated in the  $xz$  and  $yz$  planes, using a geometric integration procedure that ensures angular continuity and consistency between successive measurements Fig. 5(bottom-right). In this way, the test bench provides a robust and accurate experimental characterization of the manipulator's kinematic behavior, serving as a fundamental tool for evaluating the accuracy of the theoretical model and the performance of the proposed control strategy.

## 8. INTEGRATION AND PHYSICAL CONSTRAINTS FOR EXPERIMENTS

In implementation, tendon commands are applied in discrete time. The tendon states evolve according to

$$\mathbf{L}(k+1) = \mathbf{L}(k) + \dot{\mathbf{L}}(k) \Delta t. \quad (6)$$

To ensure hardware safety and avoid cable slack or over-tensioning, the tendon lengths and rates are saturated as

$$\mathbf{L}_{\min} \leq \mathbf{L}(k) \leq \mathbf{L}_{\max}, \quad \|\dot{\mathbf{L}}(k)\|_{\infty} \leq \dot{L}_{\max}. \quad (7)$$

These constraints preserve the feasible curvature range of the backbone and guarantee safe execution on the physical continuum manipulator.

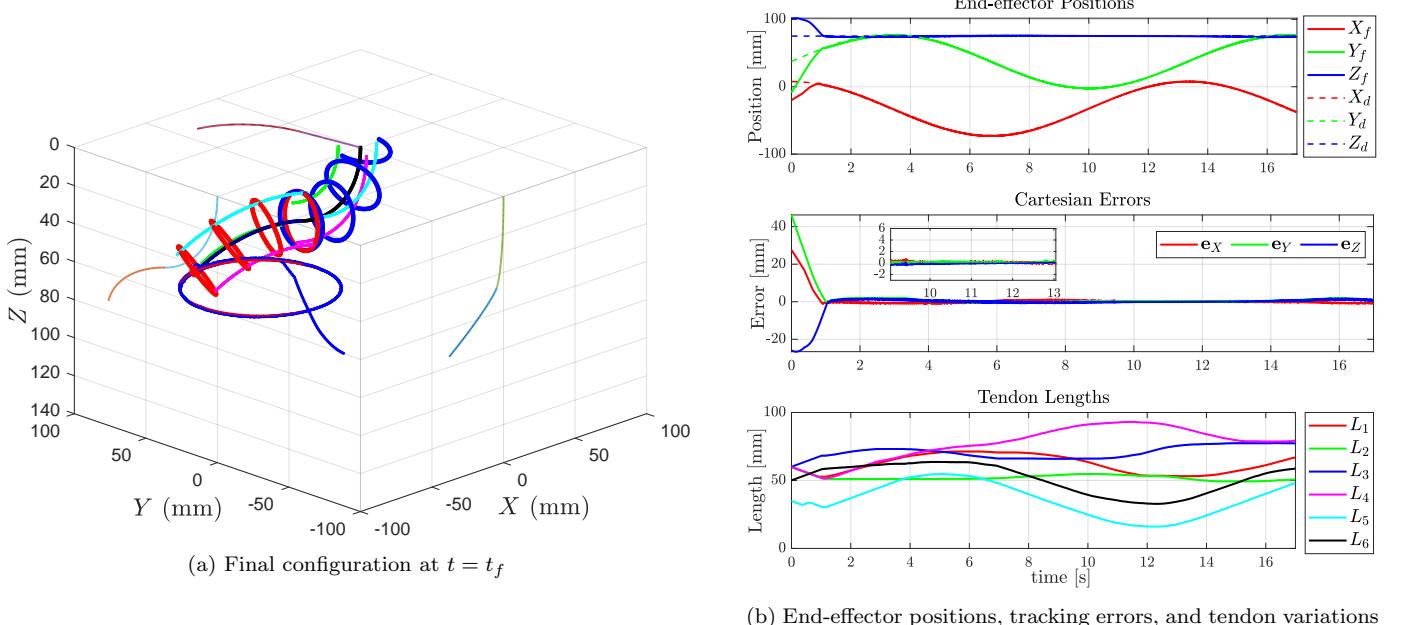


Fig. 4. Simulation results of circular trajectory tracking under the Jacobian-based control with dynamic extension. (a) shows the final spatial configuration of the soft manipulator at  $t = t_f$ , where the solid colored traces represent the backbone and segment positions, and the dashed and bold lines correspond to the desired and actual end-effector trajectories. The three faint, thin curves visible on the sides of each panel are the orthogonal projections of the desired and actual trajectories onto the XY, XZ, and YZ planes, included to provide depth perception. (b) presents the temporal evolution of the end-effector position, Cartesian tracking errors, and tendon length variations. A small residual oscillation is observed in the error signals due to numerical differentiation and discretization effects in the Jacobian update.

#### Appendix A. ANALYTICAL JACOBIAN

This appendix derives the closed-form Jacobian  $\mathbf{J} = \frac{\partial \mathbf{P}_f}{\partial \mathbf{L}}$ , which maps differential tendon length variations into end-effector motion. Although the controller in Section 6 employs the numerical Jacobian (Eq. 17) to maintain robustness when the arm departs from the constant-curvature assumption, the analytical Jacobian is included here for completeness and model validation.

**Forward kinematics of each section.** Under the constant-curvature model, the backbone of section  $j$  is parameterized by its bending direction  $\alpha_j$ , bending magnitude  $\beta_j$ , and neutral-axis arc length  $L_{cj}$ . The position contribution of section  $j$  expressed in the global frame is

$$\mathbf{p}_j(\alpha_j, \beta_j, L_{cj}) = \mathbf{R}_z(\alpha_j) \begin{bmatrix} 0 \\ \frac{L_{cj}}{\beta_j}(1 - \cos \beta_j) \\ \frac{L_{cj}}{\beta_j} \sin \beta_j \end{bmatrix}, \quad (\text{A.1})$$

where  $\mathbf{R}_z(\alpha_j)$  is the rotation aligning the bending plane.

The full manipulator consists of two serial sections; hence, the end-effector position is

$$\mathbf{P}_f = \mathbf{p}_1 + \mathbf{R}_1 \mathbf{p}_2, \quad (\text{A.2})$$

where  $\mathbf{R}_j = \mathbf{R}_z(\alpha_j) \mathbf{R}_y(\beta_j)$ .

**Tendon lengths and shape parameters.** We collect the tendon lengths into

$$\mathbf{L} = \begin{bmatrix} L_{1,1} \\ L_{1,2} \\ L_{1,3} \\ L_{2,1} \\ L_{2,2} \\ L_{2,3} \end{bmatrix} \in \mathbb{R}^6, \quad \mathbf{q} = [\alpha_1, \beta_1, L_{c1}, \alpha_2, \beta_2, L_{c2}]^\top \in \mathbb{R}^6. \quad (\text{A.3})$$

Their differential relationship follows from the chain rule:

$$\mathbf{J} = \frac{\partial \mathbf{P}_f}{\partial \mathbf{L}} = \frac{\partial \mathbf{P}_f}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{L}}. \quad (\text{A.4})$$

**Block Jacobian**  $\frac{\partial \mathbf{P}_f}{\partial \mathbf{q}}$ . Differentiating (A.2) gives, for section 1 and for section 2:

$$\begin{aligned} \frac{\partial \mathbf{P}_f}{\partial \alpha_1} &= \frac{\partial \mathbf{p}_1}{\partial \alpha_1} + \frac{\partial \mathbf{R}_1}{\partial \alpha_1} \mathbf{p}_2, & \frac{\partial \mathbf{P}_f}{\partial \alpha_2} &= \mathbf{R}_1 \frac{\partial \mathbf{p}_2}{\partial \alpha_2} \\ \frac{\partial \mathbf{P}_f}{\partial \beta_1} &= \frac{\partial \mathbf{p}_1}{\partial \beta_1} + \frac{\partial \mathbf{R}_1}{\partial \beta_1} \mathbf{p}_2, & \frac{\partial \mathbf{P}_f}{\partial \beta_2} &= \mathbf{R}_1 \frac{\partial \mathbf{p}_2}{\partial \beta_2} \\ \frac{\partial \mathbf{P}_f}{\partial L_{c1}} &= \frac{\partial \mathbf{p}_1}{\partial L_{c1}}, & \frac{\partial \mathbf{P}_f}{\partial L_{c2}} &= \mathbf{R}_1 \frac{\partial \mathbf{p}_2}{\partial L_{c2}}. \end{aligned} \quad (\text{A.5})$$

To lighten notation, define

$$a_j = \frac{\partial \mathbf{P}_f}{\partial \alpha_j}, \quad b_j = \frac{\partial \mathbf{P}_f}{\partial \beta_j}, \quad c_j = \frac{\partial \mathbf{P}_f}{\partial L_{cj}}, \quad j \in \{1, 2\}. \quad (\text{A.6})$$

Then

$$\frac{\partial \mathbf{P}_f}{\partial \mathbf{q}} = [a_1 \ b_1 \ c_1 \ a_2 \ b_2 \ c_2] \in \mathbb{R}^{3 \times 6}. \quad (\text{A.7})$$

**Derivatives of  $\mathbf{p}_j$ .** From (A.1),

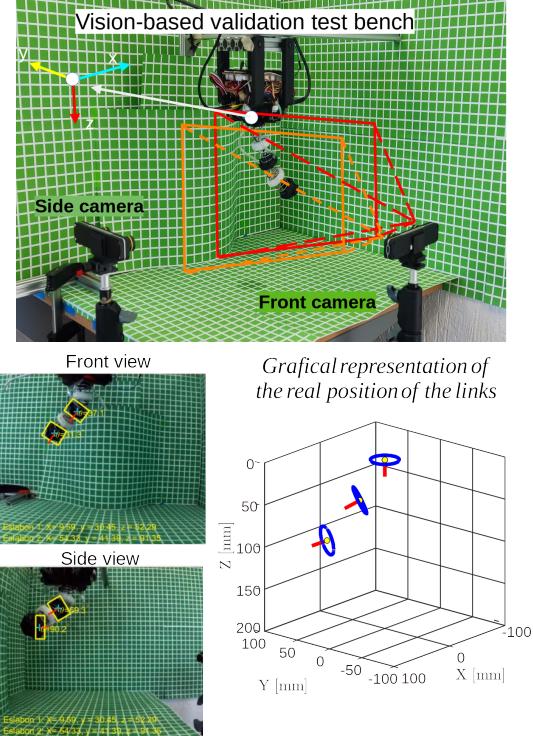


Fig. 5. Validation test bench and operation: a stereo vision system was developed to estimate the three-dimensional position and orientation of each robot segment. The top image shows the stereo vision setup; the system consists of two calibrated and synchronized stereo cameras positioned frontally and laterally. The bottom-left panel illustrates the system in operation, showing frontal and lateral camera perspectives with link centroids (cyan) and local orientations, while the bottom-right panel presents a 3D reconstruction of the link positions obtained from these images.

$$\begin{aligned} \frac{\partial \mathbf{p}_j}{\partial \alpha_j} &= \mathbf{R}_z(\alpha_j) \mathbf{S}_z \begin{bmatrix} 0 \\ \frac{L_{cj}}{\beta_j} (1 - \cos \beta_j) \\ \frac{L_{cj}}{\beta_j} \sin \beta_j \end{bmatrix} = \mathbf{S}_z \mathbf{p}_j, \\ \frac{\partial \mathbf{p}_j}{\partial \beta_j} &= \mathbf{R}_z(\alpha_j) \begin{bmatrix} 0 \\ \frac{L_{cj} \sin \beta_j - (1 - \cos \beta_j)}{\beta_j^2} \\ \frac{L_{cj} \cos \beta_j - \sin \beta_j}{\beta_j^2} \end{bmatrix}, \\ \frac{\partial \mathbf{p}_j}{\partial L_{cj}} &= \mathbf{R}_z(\alpha_j) \begin{bmatrix} 0 \\ \frac{1 - \cos \beta_j}{\beta_j} \\ \frac{\sin \beta_j}{\beta_j} \end{bmatrix} = \frac{1}{L_{cj}} \mathbf{R}_z(\alpha_j) \begin{bmatrix} 0 \\ y_j \\ z_j \end{bmatrix} = \frac{1}{L_{cj}} \mathbf{p}_j. \end{aligned} \quad (\text{A.8})$$

**Rotation derivatives.** Recall that each section orientation is

$$\mathbf{R}_j(\alpha_j, \beta_j) = \mathbf{R}_z(\alpha_j) \mathbf{R}_y(\beta_j), \quad (\text{A.9})$$

with the elementary rotations

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}. \quad (\text{A.10})$$

It is convenient to introduce the (constant) skew-symmetric generators of the  $z$ - and  $y$ -axis:

$$\mathbf{S}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [\mathbf{e}_z]_\times, \quad \mathbf{S}_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} = [\mathbf{e}_y]_\times, \quad (\text{A.11})$$

so that, for any vector  $\mathbf{v}$ ,  $\mathbf{S}_k \mathbf{v} = \mathbf{e}_k \times \mathbf{v}$  ( $k \in \{y, z\}$ ). These matrices are the Lie algebra generators of  $\text{SO}(3)$  along  $z$  and  $y$ , respectively.

*Derivative of  $\mathbf{R}_z(\alpha)$ .* From (A.10), differentiating elementwise gives

$$\begin{aligned} \frac{\partial \mathbf{R}_z(\alpha)}{\partial \alpha} &= \begin{bmatrix} -\sin \alpha & -\cos \alpha & 0 \\ \cos \alpha & -\sin \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \mathbf{R}_z(\alpha) \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{R}_z(\alpha) \mathbf{S}_z. \end{aligned} \quad (\text{A.12})$$

*Derivative of  $\mathbf{R}_y(\beta)$ .* Similarly,

$$\begin{aligned} \frac{\partial \mathbf{R}_y(\beta)}{\partial \beta} &= \begin{bmatrix} -\sin \beta & 0 & \cos \beta \\ 0 & 0 & 0 \\ -\cos \beta & 0 & -\sin \beta \end{bmatrix} \\ &= \mathbf{R}_y(\beta) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} = \mathbf{R}_y(\beta) \mathbf{S}_y. \end{aligned} \quad (\text{A.13})$$

*Derivatives of the composed rotation  $\mathbf{R}_j$ .* Using the product rule on (A.9) together with (A.12)–(A.13) yields

$$\frac{\partial \mathbf{R}_j}{\partial \alpha_j} = \frac{\partial \mathbf{R}_z(\alpha_j)}{\partial \alpha_j} \mathbf{R}_y(\beta_j) = \mathbf{R}_z(\alpha_j) \mathbf{S}_z \mathbf{R}_y(\beta_j) \quad (\text{A.14})$$

$$\frac{\partial \mathbf{R}_j}{\partial \beta_j} = \mathbf{R}_z(\alpha_j) \frac{\partial \mathbf{R}_y(\beta_j)}{\partial \beta_j} = \mathbf{R}_z(\alpha_j) \mathbf{R}_y(\beta_j) \mathbf{S}_y \quad (\text{A.15})$$

which are the compact identities used throughout the appendix.

**Dependence of  $\alpha_j, \beta_j, L_{cj}$  on tendon lengths.**

$$L_{cj} = \frac{1}{3}(L_{j,1} + L_{j,2} + L_{j,3}), \quad \frac{\partial L_{cj}}{\partial L_{j,i}} = \frac{1}{3}.$$

And since  $X_j, Y_j$  are defined as in (8), it follows that:

$$\alpha_j = \text{atan}2(Y_j, X_j), \quad \frac{\partial \alpha_j}{\partial L_{j,i}} = \frac{X_j \frac{\partial Y_j}{\partial L_{j,i}} - Y_j \frac{\partial X_j}{\partial L_{j,i}}}{X_j^2 + Y_j^2} \quad (\text{A.16})$$

and,

$$\frac{\partial X_j}{\partial L_{j,i}} = \frac{1}{3} - \cos \theta_i, \quad \frac{\partial Y_j}{\partial L_{j,i}} = \frac{1}{3} - \sin \theta_i.$$

We recall (10), where

$$\beta_j = \frac{S_{1j}}{S_{2j}}, \quad S_{1j} = \sum_{i=1}^3 p_{j,i} \Delta L_{j,i}, \quad S_{2j} = \sum_{i=1}^3 p_{j,i}^2,$$

thus,

$$\frac{\partial \beta_j}{\partial L_{j,i}} = \frac{\left( \frac{\partial S_{1j}}{\partial L_{j,i}} \right) S_{2j} - S_{1j} \left( \frac{\partial S_{2j}}{\partial L_{j,i}} \right)}{S_{2j}^2}. \quad (\text{A.17})$$

where

$$\frac{\partial S_{1j}}{\partial L_{j,i}} = p_{j,i} \left( \frac{1}{3} - 1 \right) + \sum_{k=1}^3 \Delta L_{j,k} r_j \sin(\theta_k - \alpha_j) \frac{\partial \alpha_j}{\partial L_{j,i}}, \quad (\text{A.18})$$

$$\frac{\partial S_{2j}}{\partial L_{j,i}} = 2 \sum_{k=1}^3 p_{j,k} r_j \sin(\theta_k - \alpha_j) \frac{\partial \alpha_j}{\partial L_{j,i}}. \quad (\text{A.19})$$

**Final Analytical Jacobian.** Substituting (A.7) into (A.4):

$$\mathbf{J} = \underbrace{\begin{bmatrix} a_1 & b_1 & c_1 & a_2 & b_2 & c_2 \end{bmatrix}}_{\text{Geometric Jacobian } \frac{\partial \mathbf{P}_f}{\partial \mathbf{q}}} \cdot \underbrace{\begin{bmatrix} \frac{\partial \mathbf{q}_1}{\partial \mathbf{L}_1} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}_2}{\partial \mathbf{L}_2} \end{bmatrix}}_{\text{Tendon-to-shape mapping } \frac{\partial \mathbf{q}}{\partial \mathbf{L}}}. \quad (\text{A.20})$$

where the terms  $a_j, b_j, c_j$  are defined in (A.6), and

$$\frac{\partial \mathbf{q}_j}{\partial \mathbf{L}_j} = \begin{bmatrix} \frac{\partial \alpha_j}{\partial L_{j,1}} & \frac{\partial \alpha_j}{\partial L_{j,2}} & \frac{\partial \alpha_j}{\partial L_{j,3}} \\ \frac{\partial \beta_j}{\partial L_{j,1}} & \frac{\partial \beta_j}{\partial L_{j,2}} & \frac{\partial \beta_j}{\partial L_{j,3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}, \quad j = 1, 2. \quad (\text{A.21})$$

where the terms  $\frac{\partial \alpha_j}{\partial L_{j,i}}$  are as in (A.16), and  $\frac{\partial \beta_j}{\partial L_{j,i}}$  are defined in (A.17).

The Jacobian  $\mathbf{J} \in \mathbb{R}^{3 \times 6}$  has full row rank in all non-singular configurations, i.e.,  $\text{rank}(\mathbf{J}) = 3$ . In this regime, the manipulator can generate arbitrary differential motions of the end-effector in  $\mathbb{R}^3$ , while retaining three internal degrees of redundancy. The nullspace of  $\mathbf{J}$ ,  $\ker(\mathbf{J}) \subset \mathbb{R}^6$ , corresponds to coordinated tendon motions that do not affect the end-effector pose but modify the internal shape or redistribute tendon tension.

Rank deficiency occurs when one or both sections approach zero curvature ( $\beta_j \rightarrow 0$ ) or when their bending directions become collinear ( $\alpha_1 = \alpha_2$  and  $\beta_1 = \beta_2$ ), leading to loss of independent control directions. In such singular configurations,  $\text{rank}(\mathbf{J}) < 3$ , the reachable instantaneous motion space of the end-effector collapses, and only a subset of Cartesian directions can be actuated.