CREATE Lab Semester Project

Active Control of a Continuum Robot Based on a Sequential Locking Mechanism

Louis Horak
Robotics, EPFL

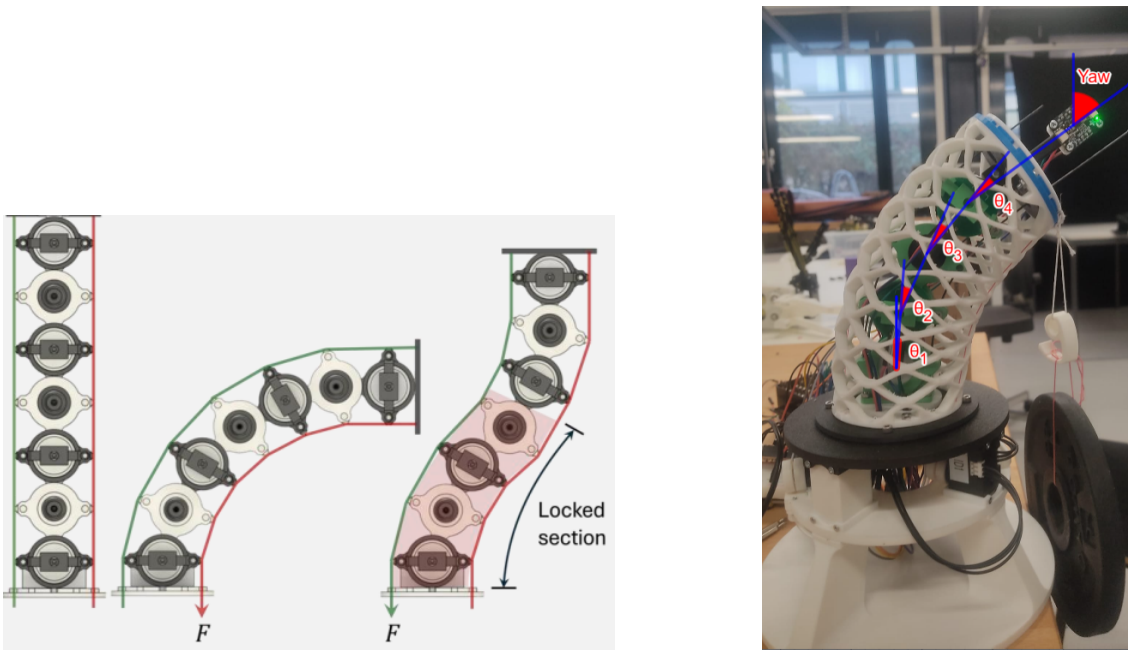
January 2026

1 Introduction

Soft continuum robots offer inherent safety and flexibility, making them ideal for applications such as minimally invasive surgery or navigation in complex environments [1]. However, they often suffer from low payload capacity and imprecise control due to their inherent compliance and complex modeling [2]. To address this trade-off, Wan introduced a hybrid robotic arm design combining a compliant "trimmed helicoid" outer structure with a rigid internal structure [3].

This internal backbone, the Lockable Spine Mechanism (LSM), utilizes a series of Hirth joints—conical teeth that interlock to prevent rotation. A single tendon actuates these joints. While the tension in the tendon pulls the faces together to ensure mechanical locking, the friction along the tendon path allows the joints to engage sequentially from base to tip [3]. This enables the robot to transition from a flexible state for navigation to a rigid state for load bearing.

However, the original work focused primarily on mechanical validation and open-loop actuation. A significant limitation of the previous prototype is the inability to control multiple curvature configurations in closed-loop. In the unlocked compliant state, the robot remains susceptible to external disturbances and hysteresis. This project aims to bridge this gap by implementing a closed-loop control strategy. By integrating distal IMU sensing and active feedback, we aim to accurately regulate the bending angle ψ, which corresponds to rotation about the yaw axis of the distal IMU (hence labeled "Yaw" in all experimental plots), and maintain a desired configuration, effectively using a sequence of single curvatures to achieve complex multi-curvature shapes despite sensing constraints.



(a) Sequential locking mechanism. **(b)** Angle convention.

Figure 1: (a) Principle of the sequential locking mechanism. (b) Angle convention for bending and yaw.

2 Methods

2.1 State Estimation Strategy

The initial control strategy aimed to utilize the four optical encoders at each joint to determine multiple curvatures. However, their coarse resolution (3.75°) and significant quantization noise made smooth feedback control impossible. While the integration of absolute PHS11 analog potentiometers was planned to resolve this, hardware delivery delays necessitated an alternative approach using the distal IMU.

To achieve multi-curvature shapes despite these constraints, a sequential control and locking strategy was implemented. The system relies on a proportional relationship between the global distal yaw ψ—measured

by the IMU's local yaw axis and corresponding to the global bending angle—and the individual joint angles θ_i: $\theta_i \approx K_i \cdot \psi$. This relationship was empirically validated through controlled bending cycles, correlating discrete encoder steps with high-resolution IMU data. An automated calibration tool sweeps the arm through its range of motion ($\pm 30^\circ$) using ramped velocity profiles to minimize inertial artifacts. A linear regression identified the coefficients K_i, allowing for continuous estimation of the robot's configuration.

The control process operates as follows (Fig. 2):

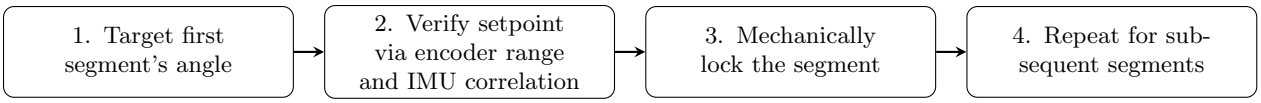


Figure 2: Sequential control and locking process for multi-curvature shapes.

The calibration yielded the following coefficients:

$$K_1 = -0.131, \quad K_2 = -0.328, \quad K_3 = -0.337, \quad K_4 = -0.198$$

2.2 System Identification

A frequency-domain analysis was performed using a chirp excitation signal (0.1 rad/s to 3 rad/s). The open-loop transfer function relates the differential velocity command u to the distal yaw ψ.

To ensure signal quality, raw IMU data were processed through a median filter to remove impulsive noise spikes. This filter introduces a frequency-dependent phase delay. This lag was modeled and mathematically included in our best-fit algorithm to estimate the transfer function. The Bode plot (Fig. 3) confirms the integrating behavior of $H(s)$. The phase response includes the explicitly modeled delay from the median filter (window size $k = 5$), which was taken into account in the controller stability analysis.

The resulting Bode analysis (Fig. 3) yielded the transfer function:

$$H(s) = \frac{-14.72}{s^2 + 4.574s} \tag{1}$$

Transfer function parameters were estimated via least-squares optimization on the frequency-domain data using a custom pole-zero fitting tool.

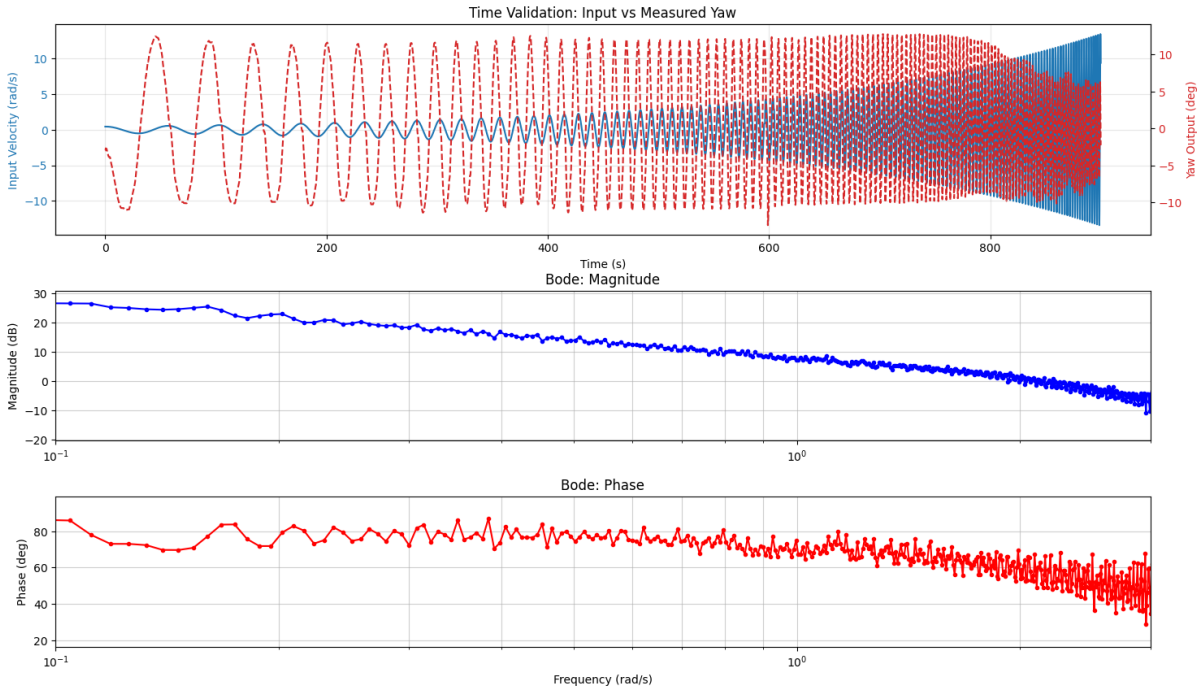


Figure 3: Bode plot of identified transfer function $H(s)$.

2.3 Controller Design and Stability Analysis

A proportional controller computes the velocity command: $u(t) = K_p(\psi_{\text{target}} - \psi_{\text{measured}})$. The negative sign of the plant $H(s)$ requires $K_p < 0$ for correct feedback polarity. The middle plot in Fig. 4 shows the control signal u in rad/s (velocity command to motors).

Despite its simplicity, a P-controller was found to be sufficient for several reasons. First, the plant $H(s)$ exhibits inherent integrating behavior (as seen in its transfer function), which theoretically ensures zero steady-state error for step inputs without the need for an integral (I) term. Second, the Dynamixel XC330 motors feature an internal hardware PID loop for velocity and position control; this low-level regulation likely contributes to the system's high stability and performance, rendering additional derivative (D) action in the high-level controller unnecessary.

Stability and Delay Compensation The median filter (window size $k = 5$, sampling frequency $f_s = 50\,\text{Hz}$) introduces a delay $\tau = \frac{k-1}{2f_s} \approx 0.04\,\text{s}$. Using a first-order approximation ($e^{-\tau s} \approx 1-\tau s$), the closed-loop characteristic equation becomes:

$$s^2 + (B + AK_p\tau)s - AK_p = 0 \tag{2}$$

where $A = 14.72$ and $B = 4.574$. Stability requires $B + AK_p\tau > 0$. With $K_p = -0.5$, the system remains stable with a predicted damping ratio $\zeta \approx 0.79$, ensuring a fast response with minimal overshoot.

2.4 Automated Performance Testing

A comprehensive GUI was developed to automate testing and data collection. The interface provides:

- Real-time plotting of yaw, control signal, and motor currents
- Automated step response and disturbance rejection tests
- Parameter adjustment and calibration procedures
- CSV export of all test data for analysis

This interface enabled efficient parameter tuning and real-time monitoring during experiments.

3 Experimental Setup

3.1 Hardware Configuration

The continuum arm is actuated by two Dynamixel XC330 motors in a differential tendon configuration. The sensing system comprises:

- **Global orientation**: BNO055 IMU mounted at the distal end
- **Joint angles**: 4× optical encoders with 3.75° resolution
- **Control interface**: U2D2 for Dynamixel communication, FTDI for IMU

3.2 Software Implementation

The control loop runs at 50 Hz in Python, with separate threads for sensor reading, motor control, and GUI updates. The median filter operates with a 5-sample window to reject noise spikes while minimizing phase delay.

3.3 Automated Calibration

The calibration procedure utilizes a velocity-ramped profile to minimize inertial effects. By sweeping the arm from 0° to 30° to −30° and back while logging IMU yaw and encoder counts, the system identifies the K_i coefficients via least-squares regression. This ensures the estimation is robust to variations in tendon tension.

4 Results

4.1 Step Response Performance

Figure 4 shows the system's response to a 20° step command. The top plot displays the measured bending angle ψ (labeled *Yaw* per IMU convention), demonstrating accurate tracking. The middle plot shows the control signal u in rad/s, and the bottom plot shows motor currents, which reflect the control effort required.

The response shows a rise time of 1.07 s and a settling time of 1.67 s. The steady-state error is $-0.11°$, demonstrating the effectiveness of the proportional controller combined with the system's integrating dynamics.

Metric	Theory ($\zeta = 0.79$)	Experimental	Error
Rise Time t_r	0.92 s	1.07 s	16.3%
Settling Time t_s	1.45 s	1.67 s	15.2%
Overshoot M_p	1.9 %	0.65 %	-65.8%

Table 1: Comparison of predicted and experimental dynamic characteristics. Values represent a typical run from a series of consistent trials with low standard deviation. The theoretical values are derived from the identified model with $K_p = -0.5$.

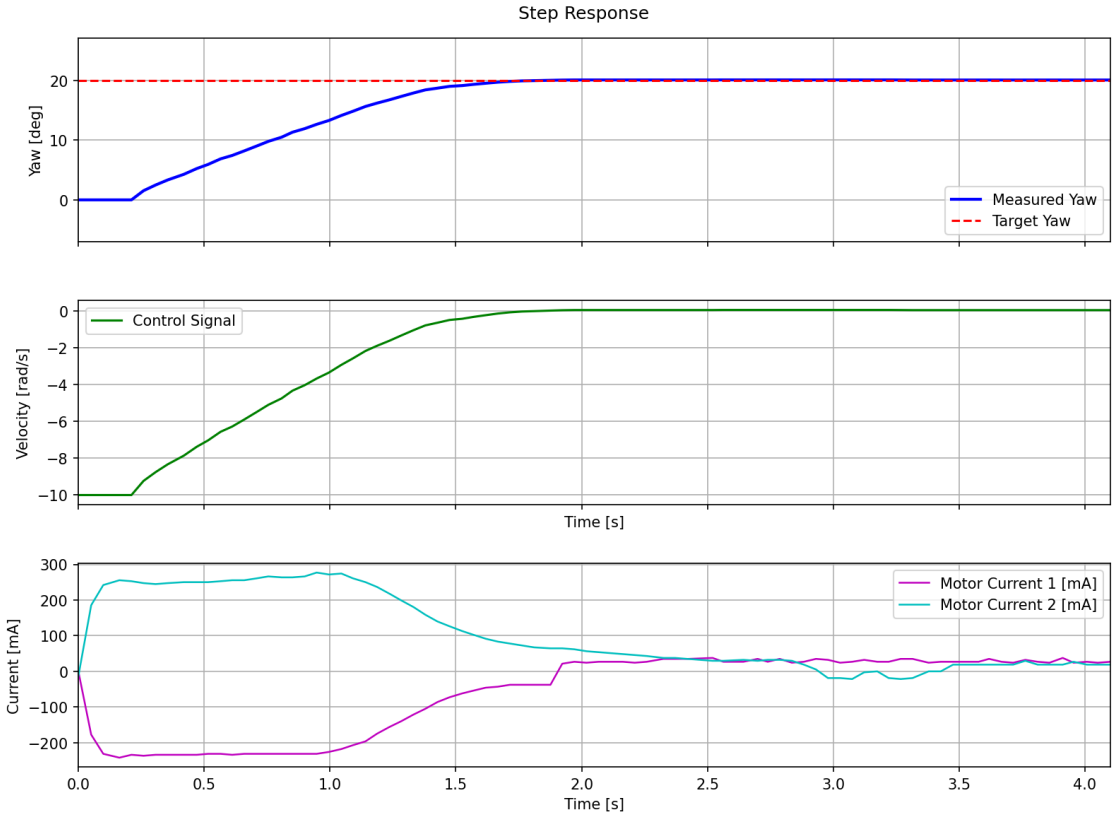


Figure 4: Step response to 20° command: bending angle (Yaw), control signal, and motor currents.

4.2 Active Disturbance Rejection

Figure 5 illustrates active disturbance rejection at $-45°$. The transient between $t = 0$ and $t = 4$s corresponds to manual placement of a 1 kg load. During this interval, the recorded deviation is due to the manual handling of the arm. Once released, the controller immediately compensates for the load, returning the system to the setpoint in approximately 1.5 s. The Dynamixel motors effectively absorb the external tension, providing "active stiffness" to the compliant structure.

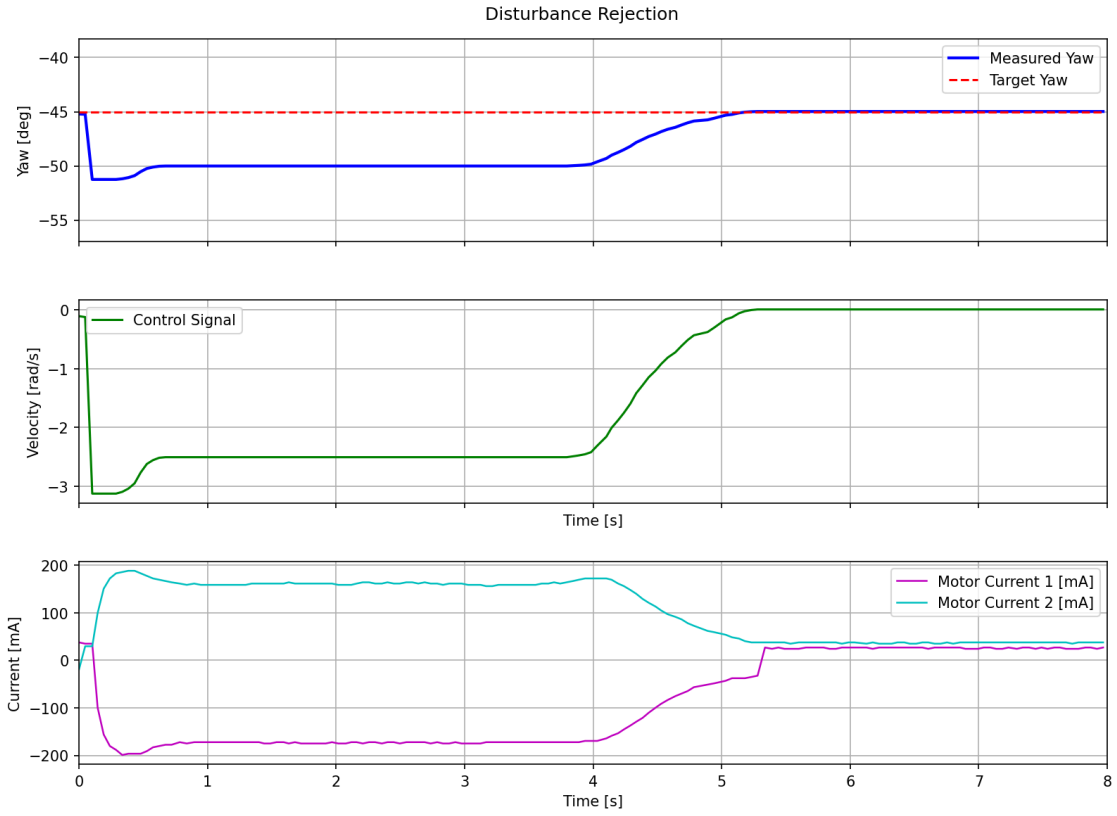


Figure 5: Disturbance rejection at -45° under 1 kg load.

5 Discussion & Conclusion

The results validate that a simple proportional controller is sufficient to stabilize the integrating dynamics of the tendon-driven system. The disturbance rejection test proves that active control can serve as a functional substitute for mechanical locking during dynamic phases, providing "active stiffness" through continuous feedback compensation.

5.1 Integration with Sequential Locking

The calibrated K_i coefficients serve a dual purpose: they enable state estimation for control and provide the kinematic mapping needed for sequential locking. When a segment is locked, the corresponding K_i effectively becomes 0, and the control law can be adjusted accordingly. This provides a straightforward pathway for integrating active control with mechanical locking operations.

5.2 Limitations and Future Improvements

The K_i-based estimation assumes uniform bending, which may not hold under complex loads. The immediate path to improvement is the integration of PHS11 analog potentiometers. These absolute sensors will provide continuous joint-angle measurements, eliminating the proportional assumption and enhancing precision. Future work includes:

- Implementing sequential locking control using the PHS11 sensor
- Extending control to 3D using full IMU orientation

5.3 Conclusion

This work successfully transitions the hybrid continuum robot from a mechanical prototype to a controlled system. The calibrated K_i coefficients now serve as a bridge between continuous active control and discrete sequential locking states, enabling precise configuration control while maintaining compliance advantages.

The system achieves sub-degree accuracy, rapid response, and effective disturbance rejection, validating the approach for hybrid continuum robot control.

References

[1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015. DOI: 10.1109/TRO.2015.2489500.

[2] I. D. Walker, "Continuous backbone "Continuum" robot manipulators," *ISRN Robotics*, vol. 2013, p. 726 506, 2013. DOI: 10.5402/2013/726506.

[3] J. Wan, "Sequential locking mechanism based on Hirth joints for continuum structures," M.S. thesis, EPFL, 2024.

Appendix: Code and CAD Resources

Source Code

All Python source code is available on GitHub at:

https://github.com/LouisH14/SEMESTER_PROJECT_CREATE_LAB

The repository contains the following files:

- **MotorWidebandID.py** (579 lines) – System identification via chirp excitation and cross-spectral density analysis
- **SequentialJointCalibration.py** (960 lines) – Automated K_i coefficient calibration with velocity-ramped profiles
- **TransferFunctionEstimator.py** (428 lines) – Frequency-domain transfer function fitting via least-squares optimization
- **robot_performance_tests.py** (1314 lines) – Comprehensive testing framework for step response and disturbance rejection
- **dynamixel_controller.py** (474 lines) – Hardware abstraction layer for Dynamixel motor control and IMU interfacing

CAD Components

The complete mechanical design is available in Autodesk Fusion 360 format. All CAD files and components are packaged in the file `ASSEMBLAGE.f3z`.

The assembly file is located in the CREATE Lab Fusion 360 workspace at: `Soft_Sensors_and_bodies > Louis_Horak > ASSEMBLAGE.f3z`. The file imports all custom components created for this project.