# Robotic Control of an Elastic Beam

Jiajin Cui

***Abstract***— **This report represents the implementation of a planer control system for manipulating a slender elastic aluminum beam.**

## I. Introduction

The objective of this homework is to design a controller such that the beam's middle node follows a prescribed trajectory with the left end remains fixed and the right end actuated by the robot end effector.

## II. Methodology

### *A. Discretization*

The beam is modeled as a discrete mass-spring system with 19 nodes. The system is described by a position vector q

### *B. Energy*

The total elastic energy is the linear sum of stretching energy and bending energy. The stretching energy for spring k connecting node k and k+1 is:

$$E_k^s = \frac{1}{2} EA\Delta L \left( \frac{\ell_k - \Delta L}{\Delta L} \right)^2$$

where l_k is the deformed length.

The bending energy at node k is:

$$E_k^b = \frac{1}{2} \frac{EI}{\Delta L} (\kappa_k - \bar{\kappa}_k)^2$$

The discrete curvature is computed using the turning angle

$$\kappa_k = \frac{2\mathbf{t}_{k-1} \times \mathbf{t}_k}{1 + \mathbf{t}_{k-1} \cdot \mathbf{t}_k}$$

where t_k-1 and t_k are unit tangent vectors of edge k-1 and k

### *C. Forces*

The only external force would be gravity on each node:

$$\mathbf{f}_g^{(i)} = \begin{bmatrix} 0 \\ -mg \end{bmatrix}, \quad g = 9.81 \text{ m/s}^2$$

### *D. Time Integration*

As usual, the equations of motion are integrated using an Implicit Euler time-stepping

$$\mathbf{M}\frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t^2} - \mathbf{M}\frac{\mathbf{u}^n}{\Delta t} = \mathbf{F}_{\text{elastic}}(\mathbf{q}^{n+1}) + \mathbf{F}_{\text{gravity}}$$

### *E. Boundary Conditions*

The left end is clamped at (0,0):

$$x_1(t_{k+1}) = 0, \quad y_1(t_{k+1}) = 0$$

The rightmost node is controlled by the robot, and using Dirichlet constraints, we have:

$$x_N(t_{k+1}) = x_c(t_{k+1}), \quad y_N(t_{k+1}) = y_c(t_{k+1})$$
$$x_{N-1}(t_{k+1}) = x_c(t_{k+1}) - \Delta L \cos\theta_c(t_{k+1})$$
$$y_{N-1}(t_{k+1}) = y_c(t_{k+1}) - \Delta L \sin\theta_c(t_{k+1})$$

### *F. Control Law*

A for-loop based simple controller is implemented. The design is to measure the error in the middle node position and adjust the inputs to reduce the error.

**Algorithm 1** Control Law for Robot End-Effector

**Input:** Current configuration $\mathbf{q}$, time $t$, current controls $(x_c, y_c, \theta_c)$
**Output:** Updated controls $(x_c^{\text{new}}, y_c^{\text{new}}, \theta_c^{\text{new}})$

Compute target position: $x^* \leftarrow \frac{L}{2}\cos\left(\frac{\pi t}{2000}\right)$, $y^* \leftarrow -\frac{L}{2}\sin\left(\frac{\pi t}{2000}\right)$

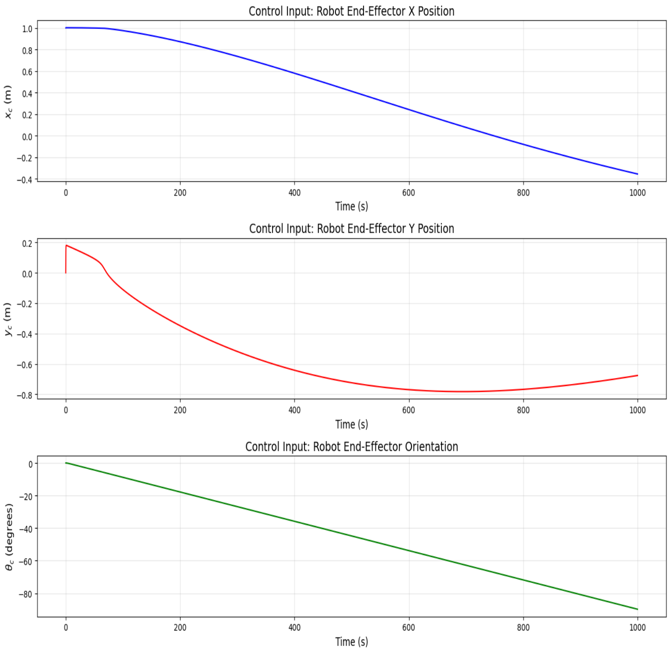
Geometric positioning: $x_c^{\text{geo}} \leftarrow 2x^*$, $y_c^{\text{geo}} \leftarrow 2y^*$

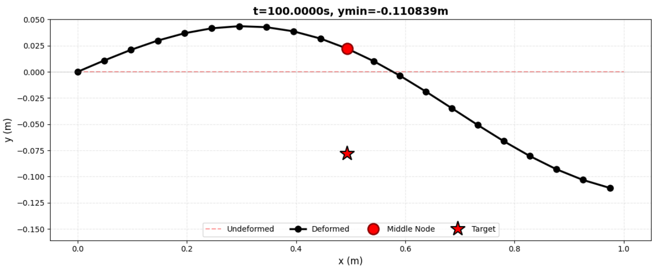
Elevation compensation: $\text{base\_lift} \leftarrow 0.15$, $\text{depth\_factor} \leftarrow 0$
**for** $i = 1$ to $10$ **do**
  $\text{depth\_factor} \leftarrow \text{depth\_factor} + |y^*|/10$
**end for**
$y_c^{\text{geo}} \leftarrow y_c^{\text{geo}} + \text{base\_lift} + 0.3 \cdot \text{depth\_factor}$

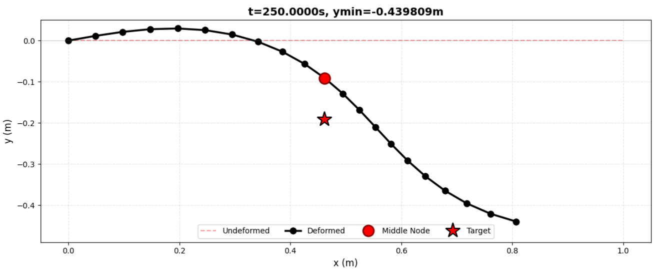
Set orientation: $\theta_c^{\text{geo}} \leftarrow \arctan 2(y^* - y_c^{\text{geo}}, x^* - x_c^{\text{geo}})$

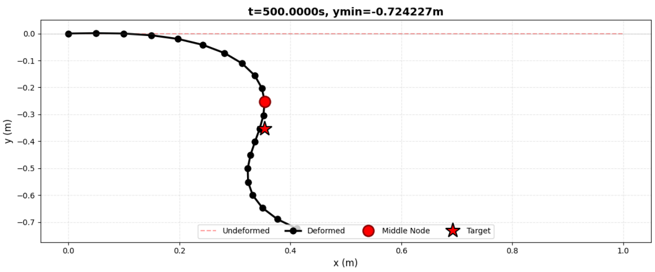
Apply smoothing $(\alpha = 0.95)$:
  $x_c^{\text{new}} \leftarrow \alpha x_c + (1-\alpha)x_c^{\text{geo}}$
  $y_c^{\text{new}} \leftarrow \alpha y_c + (1-\alpha)y_c^{\text{geo}}$
  $\theta_c^{\text{new}} \leftarrow \alpha \theta_c + (1-\alpha)\theta_c^{\text{geo}}$

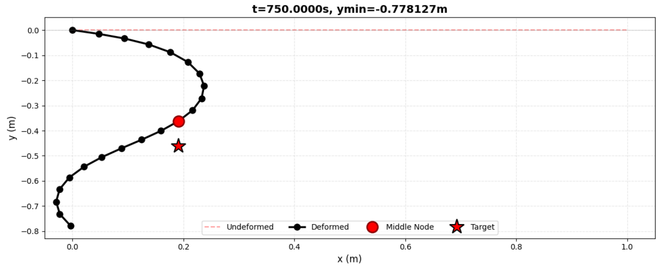
**return** $(x_c^{\text{new}}, y_c^{\text{new}}, \theta_c^{\text{new}})$

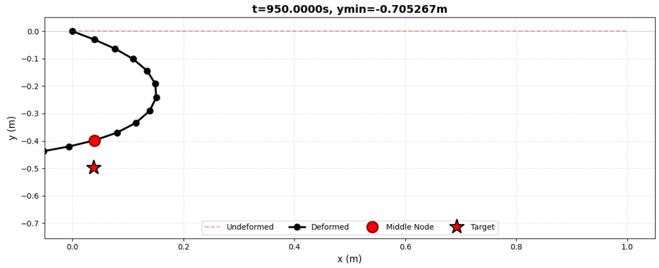



## III. Discussion

### A. *Workspace Limit*

The maximum reach and joint structure of the robotic arm may make certain path infeasible.

### B. *Joint Limit*

Typical robotic joints may not be able to rotate about 3 axes for 360, which would lead to infeasible orientation.

### C. *Infeasible Commands*

Saturation: We can manually adjust maximum allowed values

Trajectory re-timing: When approaching limits, slowing down the trajectory may help with a more stable path / motion.

## References

Professor M.Khalid Jawed, “Main_Falling_Beam_N_nodes_Fixed_DOF(Colab notebook),” Google Colab, 2025. [Online]. Available: https://colab.research.google.com/drive/1O-YT-XCm3KsTVpRGLrfuAk4S2qzCUnD1?usp=sharing&authuser=1Accessed: Nov 5, 2025.