ME130 Project Design Report: Automated Bow Maker

Oliver Chang

May 2026

1 Introduction

I have always been fascinated by vintage mechanically actuated machines. Vintage bow makers, once commonplace in department stores, convert a continuous rotation input from a crankshaft into synchronized intermittent motion, allowing the machine to sequentially pull, fold, and pin polypropylene ribbon into a multi-petal star bow.

2 Constraints

The footprint of the machine needs to be compact and portable, and the crank-rocker linkage also has to meet constraints for its sweep path, mechanical advantage, and timing. First, the folding arm's end-effector must strike the pinning platform at precisely $Y = 0$ at its extreme left stroke. The mechanism must maintain this invariant zero-point reach across all adjustable bow sizes (parameter c). Second, the mechanism must have a high enough mechanical advantage exactly at the pinning moment to maximize puncturing force. Finally, the linkage must be synthesized to achieve a quick-return time ratio ($T_R > 1$).

3 Design Methods

To satisfy spatial and timing requirements, a crank-rocker four-bar linkage paired with a system of bevel gears was selected.

3.1 Analytical Kinematic Synthesis

The mechanism was synthesized by coupling the internal linkage constraints with the required end-effector path governed by the vector loop closure equation:

$$ae^{j\theta_2} + be^{j\theta_3} - ce^{j\theta_4} - d = 0 \tag{1}$$

3.2 End-Effector Path Mapping

The spatial position R_P is defined as:

$$R_P = O_4 + L_{arm}e^{j(\theta_4+\gamma_{arm})} + L_{off}e^{j(\theta_4+\gamma_{arm}-\pi/2)} \tag{2}$$

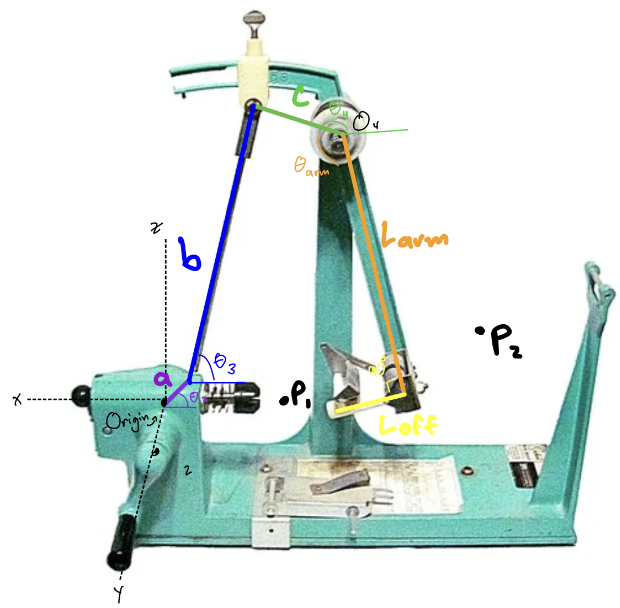


Figure 1: Labeled drawing with all linkage variables and angles.

3.3 Displacement Mapping

The displacement vector P_{21} eliminates the ground pivot O_4:

$$P_{21} = L_{arm}e^{j\gamma_{arm}} + L_{off}e^{j(\gamma_{arm}-\pi/2)}(e^{j\theta_{4,2}} - e^{j\theta_{4,1}}) \tag{3}$$

3.4 Mechanical Advantage Optimization

The linkage was optimized for MA (analysis in **Appendix A3**):

$$MA = \frac{\omega_2}{\omega_4} = \frac{c\sin(\theta_4 - \theta_3)}{a\sin(\theta_2 - \theta_3)} \tag{4}$$

3.5 Bevel Gear Transmission and Synchronization

A continuous bevel gear with $N_{driver} = 28$ and $N_{driven} = 24$ was synthesized, yielding a constant gear ratio (G_R):

$$G_R = \frac{N_{driver}}{N_{driven}} = \frac{28}{24} = 1.167 \tag{5}$$

This ratio ensures that for every full rotation of the input crank, the output indexing mechanism rotates by approximately 420°.

3.6 Kinematic and Dynamic Analysis

Following the transmission synthesis, numerical differentiation of the end-effector position (Eq. 2) was performed to determine the linear velocity and acceleration of the needle tip. The dynamic inertial impact force (F_{impact}) was then calculated based on the effective mass of the rocker assembly (m_{arm}):

$$F_{impact} = m_{arm} \cdot \frac{d^2 R_P}{dt^2} \tag{6}$$

4 Results

The analytical dyad synthesis successfully anchored the mechanism, yielding ground pivot coordinates of $O_2 = (0, 0)$ mm and $O_4 = (150, 250)$ mm. Subsequently, the computational optimization yielded a required bell crank angle of 112.8°. As shown in Figure 2, the generated spatial envelope confirms the end-effector accurately bottoms out at exactly $Y = 0$ mm.

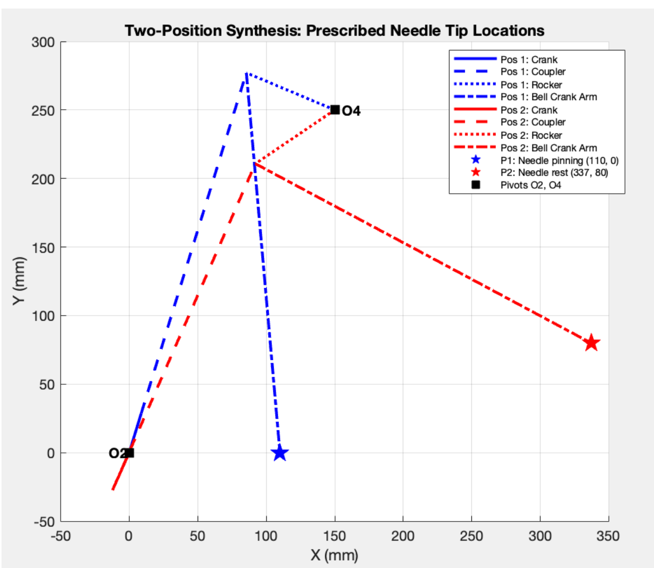


Figure 2: 2D Spatial Path confirming the two prescribed positions and stroke constraint at Y=0.

To evaluate the adjustability of the machine for various bow sizes, parameter sweeps were conducted for the slot adjuster length (c) and the needle offset (L_{offset}). As illustrated in Figure 7 (Appendix A4), varying the slot length alters the vertical and horizontal extent of the stroke, allowing the machine to accommodate different ribbon looping diameters. Modifying the needle offset primarily shifts the horizontal reach of the spatial envelope. Notably, the critical pinning constraint ($Y = 0$ at the leftmost stroke) is strictly preserved across these adjustments.

Kinematic analysis confirmed the machine operates efficiently and safely. The Mechanical Advantage spikes precisely when the crank angle reaches the pinning phase, ensuring maximum leverage when puncturing the ribbon (see Figure 7a, Appendix A3).

Dynamic force analysis calculated a peak inertial impact force of approximately 0.73 N at the needle tip, confirming the mechanism provides sufficient force to puncture the ribbon.

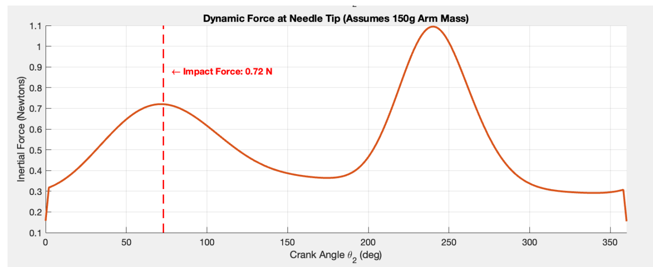


Figure 3: Dynamic Impact Force across one full rotation.

The final physical dimensions of the prototype are: Input Crank (a): 30 mm, Coupler (b): 260 mm, Slot Adjuster (c): 70 mm, Pendulum Arm: 250 mm.

5 Discussion

While the linkage achieved synchronized motion, physical ribbon folding failed. Future iterations must integrate a tensioning feed-spool and mechanical stripper plate.

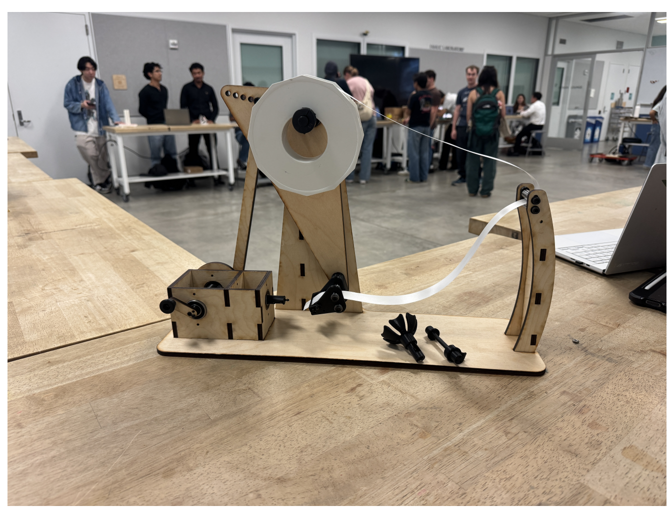


Figure 4: Final fabricated physical prototype.

Appendix

A1. Physical Prototype Fabrication Details

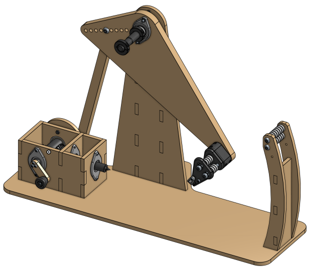
Item	**Material**	**Source**	**Cost ($)**
Structure	1/4" Plywood	Jacobs Hall	10.00
Linkages	1/4" Plywood	Jacobs Hall	0.00
Bearings & Hardware	Aluminum/Steel	Amazon	10.00
Ribbon & Hardware	Polypropylene	Amazon	10.00
Gears	PLA (3D Printed)	Jacobs Hall	0.00
Total			**30.00**

A2. Graphical Synthesis

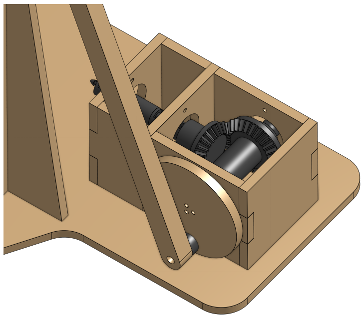
Figure 5: Hybrid graphical synthesis for end-effector optimization.

Graphical Synthesis: For graphical synthesis I used a combination of timing 2 point synthesis to locate my unknown variables, which were the linkage c length, angle, and O4 grounding point. The crank a and coupler b established by timing constraints, the unknown ground pivot O_4 and rocker length c are derived by constructing the perpendicular bisector of the needle tip displacement chord $\mathbf{P}_{21}$. This geometric construction ensures the needle tip arc passes through the prescribed active and rest positions, effectively synthesizing link c to accommodate specific ribbon dimensions.

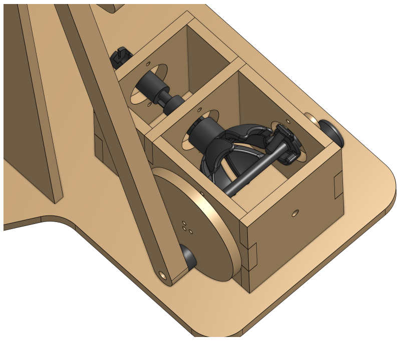
A3. CAD and Transmission Subsystems



(a) Full Assembly



(b) Bevel Gear Set

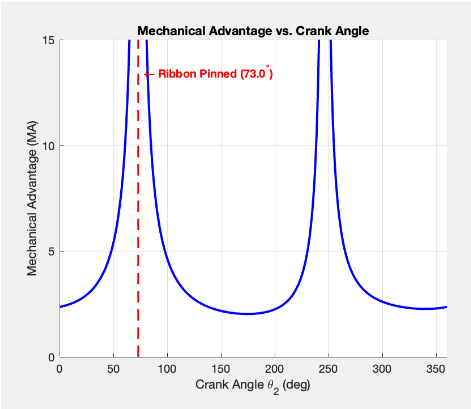


(c) Intermittent Drive

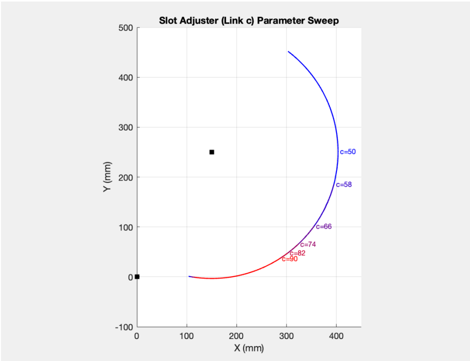
Figure 6: CAD models for the primary assembly and mechanical transmission components.

Transmission Design Evolution: While a Geneva drive (Fig. 6c) was initially considered for intermittent indexing, it was rejected due to synchronization requirements. To achieve the necessary $1\frac{1}{6}$ rotations per fold, the Geneva system would have required an additional intermediary gear to match the 1.167 ratio. The bevel gear set (Fig. 6b) was chosen instead to directly provide the required timing in a more compact, single-stage transmission.

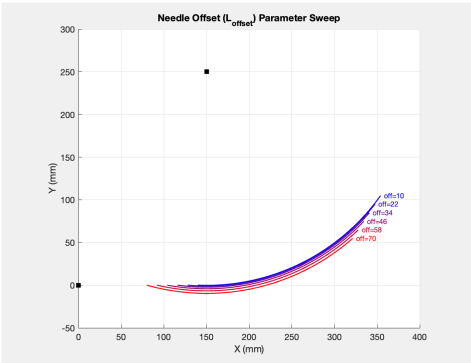
A4. Performance Metrics and Parameter Sweeps



(a) Mechanical Advantage Profile



(b) Slot Adjuster (c) Sweep



(c) Needle Offset (L_{offset}) Sweep

Figure 7: Kinematic metrics and spatial robustness analysis.

A5. MATLAB Code

```
clear; clc; close all;

% Section 1: Kinematic Synthesis
% Solve for O2 and O4 from two prescribed coupler positions
fprintf('=== KINEMATIC SYNTHESIS ===\n\n');

% Prescribed coupler positions (from needle tip requirements)
P1_needle = [110;   0    ];   % mm, pinning position
P2_needle = [337.5; 79.8 ];   % mm, rest position

% Prescribed link lengths (design choices)
a = 30;     % crank length [mm]
b = 260;    % coupler length [mm]
c = 70;     % rocker length [mm]
L_arm = 250;   % bell crank arm length [mm]
L_off = 40;    % needle offset (perpendicular to arm) [mm]

% Coupler rotation between the two positions
alpha_2 = deg2rad(66.37 - 72.84);   % = -6.47 deg (coupler rotates slightly CW)

fprintf('Prescribed two-position path generation:\n');
fprintf('  P1 (needle at ribbon): [%.1f, %.1f] mm\n', P1_needle(1), P1_needle(2));
fprintf('  P2 (needle at rest):   [%.1f, %.1f] mm\n', P2_needle(1), P2_needle(2));
fprintf('  Coupler rotation alpha_2 = %.4f deg\n\n', rad2deg(alpha_2));

% Left dyad     finds crank pivot O2
theta_pos1 = deg2rad(73);     % crank angle at position 1 [rad]
beta_2     = deg2rad(173);    % crank rotation from pos1 to pos2 [rad]

A1 = a * exp(1i * theta_pos1);
A2 = a * exp(1i * (theta_pos1 + beta_2));
p21_A = A2 - A1;

LHS_left = a * exp(1i*theta_pos1) * (1 - exp(1i*beta_2));
RHS_left  = -p21_A;

fprintf('Left Dyad Synthesis (finds O2):\n');
fprintf('  w       = %g mm  (crank length)\n', a);
fprintf('  theta   = %.2f deg  (crank angle at position 1)\n', rad2deg(theta_pos1));
fprintf('  beta_2  = %.2f deg  (crank rotation pos1 -> pos2)\n', rad2deg(beta_2));
fprintf('  z       = 0  (coupler point chosen as joint A)\n');
fprintf('  Equation check  LHS - RHS = %.2e + %.2ej\n', ...
        real(LHS_left - RHS_left), imag(LHS_left - RHS_left));
fprintf('  => O2 = (0, 0) mm  [by construction]\n\n');
X_O2 = 0;  Y_O2 = 0;

% Right dyad     finds rocker pivot O4
gamma_1 = deg2rad(157.21);   % rocker angle at position 1 [rad]
gamma_2 = deg2rad(56.85);    % rocker rotation pos1 -> pos2 [rad]

B1_complex = c * exp(1i * gamma_1);
B2_complex = c * exp(1i * (gamma_1 + gamma_2));
p21_B = B2_complex - B1_complex;

LHS_right = c * exp(1i*gamma_1) * (1 - exp(1i*gamma_2));
RHS_right  = -p21_B;

theta3_pos1 = deg2rad(72.84);
O4_complex = (X_O2 + a*exp(1i*theta_pos1) + b*exp(1i*theta3_pos1)) - c*exp(1i*gamma_1);
X_O4 = real(O4_complex);
Y_O4 = imag(O4_complex);

fprintf('Right Dyad Synthesis (finds O4):\n');
fprintf('  u       = %g mm  (rocker length)\n', c);
fprintf('  gamma_1 = %.2f deg  (rocker angle at position 1)\n', rad2deg(gamma_1));
fprintf('  gamma_2 = %.2f deg  (rocker rotation pos1 -> pos2)\n', rad2deg(gamma_2));
fprintf('  s       = 0  (coupler point chosen as joint B)\n');
fprintf('  Equation check  LHS - RHS = %.2e + %.2ej\n', ...
        real(LHS_right - RHS_right), imag(LHS_right - RHS_right));
fprintf('  => O4 = (%.2f, %.2f) mm\n\n', X_O4, Y_O4);
```

```matlab

% Bell crank angle     enforcing needle tip Y = 0 at pinning position
theta2_full = deg2rad(0:1:360);
N = length(theta2_full);
options = optimoptions('fsolve', 'Display', 'none');

getLeftY = @(gamma) min_y_at_min_x(a, b, c, X_O4, Y_O4, ...
                                    L_arm, L_off, gamma, theta2_full, N, options);
gamma_arm = fzero(getLeftY, deg2rad(112.8));

fprintf('Bell Crank Angle Synthesis:\n');
fprintf('  Constraint: needle tip Y = 0 at pinning position\n');
fprintf('  => gamma_arm = %.4f deg\n\n', rad2deg(gamma_arm));

fprintf('=== SYNTHESIS RESULTS SUMMARY ===\n');
fprintf('  O2 = (%.2f, %.2f) mm\n', X_O2, Y_O2);
fprintf('  O4 = (%.2f, %.2f) mm\n', X_O4, Y_O4);
fprintf('  Crank   a = %g mm\n', a);
fprintf('  Coupler b = %g mm\n', b);
fprintf('  Rocker  c = %g mm\n', c);
fprintf('  Bell crank arm    L_arm = %g mm\n', L_arm);
fprintf('  Needle offset     L_off = %g mm\n', L_off);
fprintf('  Bell crank angle  gamma_arm = %.4f deg\n\n', rad2deg(gamma_arm));


% Section 2: Kinematic Analysis
fprintf('=== KINEMATIC ANALYSIS ===\n\n');
[Px_perf, Py_perf, theta3_perf, theta4_perf] = calculatePath( ...
    a, b, c, X_O4, Y_O4, L_arm, L_off, gamma_arm, theta2_full, N, options);

[~, pin_idx] = min(Px_perf);
pin_angle_deg = rad2deg(theta2_full(pin_idx));

fprintf('Needle pinning event:\n');
fprintf('  At theta2 = %.1f deg\n', pin_angle_deg);
fprintf('  Needle tip = (%.4f, %.4f) mm  [Y should be ~0]\n\n', ...
        Px_perf(pin_idx), Py_perf(pin_idx));

% --- REDUCED SWEEP DATA POINTS FOR CLARITY ---
num_steps = 6; % Reduced from 20 to 6
grad_colors = [linspace(0,1,num_steps)', zeros(num_steps,1), linspace(1,0,num_steps)'];


% Section 3: Synthesis Diagram
% Shows the linkage at both prescribed positions
f0 = figure('Name', 'Synthesis: Two Prescribed Positions');
ax0 = axes('Parent', f0, 'Interactions', []);
ax0.NextPlot = 'add'; ax0.XGrid = 'on'; ax0.YGrid = 'on';
ax0.DataAspectRatio = [1 1 1];
ax0.Title.String = 'Two-Position Synthesis: Prescribed Needle Tip Locations';
ax0.XLabel.String = 'X (mm)'; ax0.YLabel.String = 'Y (mm)';

% Draw linkage at position 1 (pinning)
theta3_p1 = deg2rad(72.84);
theta4_p1 = gamma_1;

A1_plot = [a*cos(theta_pos1), a*sin(theta_pos1)];
B1_plot = [X_O4 + c*cos(theta4_p1), Y_O4 + c*sin(theta4_p1)];
aa1 = theta4_p1 + gamma_arm;
Needle1 = [X_O4 + L_arm*cos(aa1) + L_off*cos(aa1-pi/2), ...
           Y_O4 + L_arm*sin(aa1) + L_off*sin(aa1-pi/2)];

% Draw linkage at position 2 (rest)
theta2_p2 = theta_pos1 + beta_2;
theta3_p2 = deg2rad(66.37);
theta4_p2 = gamma_1 + gamma_2;

A2_plot = [a*cos(theta2_p2), a*sin(theta2_p2)];
B2_plot = [X_O4 + c*cos(theta4_p2), Y_O4 + c*sin(theta4_p2)];
aa2 = theta4_p2 + gamma_arm;
Needle2 = [X_O4 + L_arm*cos(aa2) + L_off*cos(aa2-pi/2), ...
           Y_O4 + L_arm*sin(aa2) + L_off*sin(aa2-pi/2)];
```

```
% Position 1 links (blue)
line('Parent',ax0,'XData',[X_O2, A1_plot(1)],'YData',[Y_O2, A1_plot(2)],...
     'Color','b','LineWidth',2,'DisplayName','Pos 1: Crank');
line('Parent',ax0,'XData',[A1_plot(1), B1_plot(1)],'YData',[A1_plot(2), B1_plot(2)],...
     'Color','b','LineWidth',2,'LineStyle','--','DisplayName','Pos 1: Coupler');
line('Parent',ax0,'XData',[X_O4, B1_plot(1)],'YData',[Y_O4, B1_plot(2)],...
     'Color','b','LineWidth',2,'LineStyle',':','DisplayName','Pos 1: Rocker');
line('Parent',ax0,'XData',[B1_plot(1), Needle1(1)],'YData',[B1_plot(2), Needle1(2)],...
     'Color','b','LineWidth',2,'LineStyle','-.','DisplayName','Pos 1: Bell Crank Arm');

% Position 2 links (red)
line('Parent',ax0,'XData',[X_O2, A2_plot(1)],'YData',[Y_O2, A2_plot(2)],...
     'Color','r','LineWidth',2,'DisplayName','Pos 2: Crank');
line('Parent',ax0,'XData',[A2_plot(1), B2_plot(1)],'YData',[A2_plot(2), B2_plot(2)],...
     'Color','r','LineWidth',2,'LineStyle','--','DisplayName','Pos 2: Coupler');
line('Parent',ax0,'XData',[X_O4, B2_plot(1)],'YData',[Y_O4, B2_plot(2)],...
     'Color','r','LineWidth',2,'LineStyle',':','DisplayName','Pos 2: Rocker');
line('Parent',ax0,'XData',[B2_plot(1), Needle2(1)],'YData',[B2_plot(2), Needle2(2)],...
     'Color','r','LineWidth',2,'LineStyle','-.','DisplayName','Pos 2: Bell Crank Arm');

% Prescribed needle tip points
line('Parent',ax0,'XData',Needle1(1),'YData',Needle1(2),...
     'Marker','p','MarkerSize',14,'Color','b','MarkerFaceColor','b','LineStyle','none',...
     'DisplayName',sprintf('P1: Needle pinning (%.0f, %.0f)',Needle1(1),Needle1(2)));
line('Parent',ax0,'XData',Needle2(1),'YData',Needle2(2),...
     'Marker','p','MarkerSize',14,'Color','r','MarkerFaceColor','r','LineStyle','none',...
     'DisplayName',sprintf('P2: Needle rest (%.0f, %.0f)',Needle2(1),Needle2(2)));

% Pivot markers
line('Parent',ax0,'XData',[X_O2, X_O4],'YData',[Y_O2, Y_O4],...
     'LineStyle','none','Marker','s','Color','k','MarkerFaceColor','k','MarkerSize',8,...
     'DisplayName','Pivots O2, O4');
text(X_O2-15, Y_O2, 'O2','Parent',ax0,'FontWeight','bold');
text(X_O4+5,  Y_O4, 'O4','Parent',ax0,'FontWeight','bold');
legend(ax0,'Location','best','FontSize',7);


% Section 4: Performance Plots

% Plot 1: Spatial path vs bell crank angle sweep
f1 = figure('Name', 'Plot 1: Bell Crank Angle Sweep');
ax1 = axes('Parent', f1, 'Interactions', []);
ax1.NextPlot = 'add'; ax1.XGrid = 'on'; ax1.YGrid = 'on';
ax1.DataAspectRatio = [1 1 1];
ax1.Title.String = 'Spatial Path vs. Bell Crank Angle (\gamma)';
ax1.XLabel.String = 'X (mm)'; ax1.YLabel.String = 'Y (mm)';

gamma_sweep = linspace(0, pi, num_steps);
for k = 1:num_steps
    [Px, Py, ~, ~] = calculatePath(a, b, c, X_O4, Y_O4, L_arm, L_off, ...
                                   gamma_sweep(k), theta2_full, N, options);
    line('Parent', ax1, 'XData', Px, 'YData', Py, ...
         'Color', grad_colors(k,:), 'LineWidth', 1.2);

    % Label at the rightmost point of each curve
    [~, end_idx] = max(Px);
    text(Px(end_idx) + 4, Py(end_idx), ...
         sprintf('%.0f  ', rad2deg(gamma_sweep(k))), ...
         'Parent', ax1, 'FontSize', 8, 'Color', grad_colors(k,:), ...
         'VerticalAlignment', 'middle');
end

% Synthesized path     bold black with dot marker and label
line('Parent', ax1, 'XData', Px_perf, 'YData', Py_perf, ...
     'Color', 'k', 'LineWidth', 2.5);
[~, end_idx_perf] = max(Px_perf);
text(Px_perf(end_idx_perf) + 4, Py_perf(end_idx_perf), ...
     sprintf('%.0f   (synthesized)', rad2deg(gamma_arm)), ...
     'Parent', ax1, 'FontSize', 8, 'Color', 'k', ...
     'FontWeight', 'bold', 'VerticalAlignment', 'middle');
```

```matlab
% Pivot markers
line('Parent', ax1, 'XData', [X_O2, X_O4], 'YData', [Y_O2, Y_O4], ...
     'LineStyle', 'none', 'Marker', 's', 'Color', 'k', ...
     'MarkerFaceColor', 'k', 'MarkerSize', 6);


% Plot 2: Mechanical Advantage
MA = abs((c .* sin(theta4_perf - theta3_perf)) ./ ...
         (a .* sin(theta2_full - theta3_perf)));
f2 = figure('Name', 'Plot 2: Mechanical Advantage');
ax2 = axes('Parent', f2, 'Interactions', []);
ax2.NextPlot = 'add'; ax2.XGrid = 'on'; ax2.YGrid = 'on';
ax2.Title.String = 'Mechanical Advantage vs. Crank Angle';
ax2.XLabel.String = 'Crank Angle \theta_2 (deg)';
ax2.YLabel.String = 'Mechanical Advantage (MA)';
ax2.XLim = [0 360]; ax2.YLim = [0 15];

line('Parent', ax2, 'XData', rad2deg(theta2_full), 'YData', MA, ...
     'Color', 'b', 'LineWidth', 2);
line('Parent', ax2, 'XData', [pin_angle_deg, pin_angle_deg], ...
     'YData', ax2.YLim, 'Color', 'r', 'LineStyle', '--', 'LineWidth', 1.5);
text('Parent', ax2, 'Position', [pin_angle_deg+5, 13.5], ...
     'String', sprintf('\\leftarrow Ribbon Pinned (%.1f^\\circ)', pin_angle_deg), ...
     'Color', 'r', 'FontWeight', 'bold');


% Plot 3: Dynamic Force at Needle Tip
omega2  = 2*pi;   % 60 RPM [rad/s]
dt      = (theta2_full(2) - theta2_full(1)) / omega2;
m_arm   = 0.150;  % kg

Vx = gradient(Px_perf, dt);
Vy = gradient(Py_perf, dt);
Ax = gradient(Vx, dt);
Ay = gradient(Vy, dt);
A_mag     = sqrt(Ax.^2 + Ay.^2);
F_dynamic = m_arm .* (A_mag / 1000);   % N

f3 = figure('Name', 'Plot 3: Dynamic Force');
ax3 = axes('Parent', f3, 'Interactions', []);
ax3.NextPlot = 'add'; ax3.XGrid = 'on'; ax3.YGrid = 'on';
ax3.Title.String = 'Dynamic Force at Needle Tip';
ax3.XLabel.String = 'Crank Angle \theta_2 (deg)';
ax3.YLabel.String = 'Inertial Force (N)';
ax3.XLim = [0 360];

line('Parent', ax3, 'XData', rad2deg(theta2_full), 'YData', F_dynamic, ...
     'Color', '#D95319', 'LineWidth', 2);
line('Parent', ax3, 'XData', [pin_angle_deg, pin_angle_deg], ...
     'YData', [0, max(F_dynamic)*1.1], 'Color', 'r', 'LineStyle', '--', 'LineWidth', 1.5);
text('Parent', ax3, 'Position', [pin_angle_deg+5, max(F_dynamic)*0.8], ...
     'String', sprintf('\\leftarrow Impact Force: %.2f N', F_dynamic(pin_idx)), ...
     'Color', 'r', 'FontWeight', 'bold');


% Plot 4: Link c (rocker/slot adjuster) sweep
f4 = figure('Name', 'Plot 4: Link c Sweep');
ax4 = axes('Parent', f4, 'Interactions', []);
ax4.NextPlot = 'add'; ax4.XGrid = 'on'; ax4.YGrid = 'on';
ax4.DataAspectRatio = [1 1 1];
ax4.Title.String = 'Slot Adjuster (Link c) Parameter Sweep';
ax4.XLabel.String = 'X (mm)'; ax4.YLabel.String = 'Y (mm)';

c_sweep = linspace(50, 90, num_steps);
for k = 1:num_steps
    [Px, Py, ~, ~] = calculatePath(a, b, c_sweep(k), X_O4, Y_O4, L_arm, L_off, ...
                                   gamma_arm, theta2_full, N, options);
    line('Parent', ax4, 'XData', Px, 'YData', Py, ...
         'Color', grad_colors(k,:), 'LineWidth', 1.2);

    % Directly label the rightmost point of the sweep line
    [~, end_idx] = max(Px);
```

```
    text(Px(end_idx) + 4, Py(end_idx), ...
         sprintf('c=%.0f', c_sweep(k)), ...
         'Parent', ax4, 'FontSize', 8, 'Color', grad_colors(k,:), ...
         'VerticalAlignment', 'middle');
end

line('Parent', ax4, 'XData', [X_02, X_04], 'YData', [Y_02, Y_04], ...
     'LineStyle', 'none', 'Marker', 's', 'Color', 'k', ...
     'MarkerFaceColor', 'k', 'MarkerSize', 6);


% Plot 5: Needle offset sweep
f5 = figure('Name', 'Plot 5: Needle Offset Sweep');
ax5 = axes('Parent', f5, 'Interactions', []);
ax5.NextPlot = 'add'; ax5.XGrid = 'on'; ax5.YGrid = 'on';
ax5.DataAspectRatio = [1 1 1];
ax5.Title.String = 'Needle Offset (L_{offset}) Parameter Sweep';
ax5.XLabel.String = 'X (mm)'; ax5.YLabel.String = 'Y (mm)';

off_sweep = linspace(10, 70, num_steps);
for k = 1:num_steps
    [Px, Py, ~, ~] = calculatePath(a, b, c, X_04, Y_04, L_arm, off_sweep(k), ...
                                    gamma_arm, theta2_full, N, options);
    line('Parent', ax5, 'XData', Px, 'YData', Py, ...
         'Color', grad_colors(k,:), 'LineWidth', 1.2);

    % Directly label the rightmost point of the sweep line
    [~, end_idx] = max(Px);
    text(Px(end_idx) + 4, Py(end_idx), ...
         sprintf('off=%.0f', off_sweep(k)), ...
         'Parent', ax5, 'FontSize', 8, 'Color', grad_colors(k,:), ...
         'VerticalAlignment', 'middle');
end

line('Parent', ax5, 'XData', [X_02, X_04], 'YData', [Y_02, Y_04], ...
     'LineStyle', 'none', 'Marker', 's', 'Color', 'k', ...
     'MarkerFaceColor', 'k', 'MarkerSize', 6);

fprintf('All plots generated.\n');

% Helper Functions
function [Px, Py, theta3, theta4] = calculatePath(a, b, c, X_04, Y_04, ...
        L_arm, L_offset, gamma_arm, theta2, N, options)

    theta3 = zeros(1,N);
    theta4 = zeros(1,N);
    Px     = zeros(1,N);
    Py     = zeros(1,N);

    guess  = [deg2rad(90), deg2rad(180)];

    for idx = 1:N
        eqns = @(x) [
            a*cos(theta2(idx)) + b*cos(x(1)) - c*cos(x(2)) - X_04;
            a*sin(theta2(idx)) + b*sin(x(1)) - c*sin(x(2)) - Y_04
        ];

        sol         = fsolve(eqns, guess, options);
        theta3(idx) = sol(1);
        theta4(idx) = sol(2);
        guess       = sol;

        arm_angle = theta4(idx) + gamma_arm;
        Px(idx) = X_04 + L_arm*cos(arm_angle) + L_offset*cos(arm_angle - pi/2);
        Py(idx) = Y_04 + L_arm*sin(arm_angle) + L_offset*sin(arm_angle - pi/2);
    end
end

function left_y = min_y_at_min_x(a, b, c, X_04, Y_04, ...
        L_arm, L_offset, gamma_arm, theta2, N, options)

    [Px, Py, ~, ~] = calculatePath(a, b, c, X_04, Y_04, L_arm, L_offset, ...
```

```
                                    gamma_arm, theta2, N, options);
    [~, left_idx] = min(Px);
    left_y = Py(left_idx);
end
```

Listing 1: Kinematic Synthesis and Analysis Script