

AC 2007-2190: ANALYTICAL SYNTHESIS AND ANALYSIS OF MECHANISMS USING MATLAB AND SIMULINK

Ali Mohammadzadeh, Grand Valley State University

ALI R. MOHAMMADZADEH is currently assistant professor of engineering at Padnos School of Engineering at Grand Valley State University. He received his B.S. in Mechanical Engineering from Sharif University of Technology And his M.S. and Ph.D. both in Mechanical Engineering from the University of Michigan at Ann Arbor. His research area of interest is fluid-structure interaction.

Analytical Synthesis and Analysis of Mechanisms Using MATLAB and SIMULINK

**Alireza Mohammadzadeh
Grand Valley State University**

Abstract

The approach adopted in this work is an attempt to introduce students, in kinematics and dynamics of machinery course, to a complete design and analysis of function generation mechanisms via analytical methods. Although the approach implemented in this work is for function generation type of mechanisms, the concept is indeed extendable to the other types of mechanisms as well. As a project in the kinematics and dynamics of machinery class, students designed, and analyzed a four bar quick-return mechanism using MATLAB and SIMULINK as the primary software tools. One of the aims of this project was to abandon the traditional graphical synthesis and graphical analysis, covered in all the mechanisms textbooks, and to use the powerful combination of MATLAB and SIMULINK to implement the entire design and analysis process. The project, given to an undergraduate class, serves also as a prologue to future advanced courses in mechanical engineering, such as multi-body dynamics.

In implementing the dimensional synthesis portion of the project, students employed complex number arithmetic to realize the design specifications. Once the design specifications were met, a known motor torque was applied to the crank to drive the mechanism. With the known geometric and inertial properties of each link, Lagrange's equations for constraint motion were then utilized to arrive at the second order differential equations of motion. SIMULINK, as a user friendly graphical interface, was used to carry out the integration to obtain angular position, velocity, and acceleration of the designed mechanism.

The project, though rigorous, is an excellent way to force students to practice their knowledge of dynamics and numerical methods. The project, certainly, meets the ABET criteria for implementing design in mechanical engineering curriculum. The author received positive feedbacks from his students with regard to this project.

Problem Statement

Students in kinematics and dynamics of machinery class were, first, asked to design a four bar quick-return mechanism to meet a certain design specifications. Once the mechanism was realized, students were then asked to analyze the synthesized mechanism- using Lagrange's equations and Lagrange's multipliers method for

constraint motion- to arrive at the positions, velocities and accelerations of the links. The approach, instructed to the students and utilized by them in implementing this project, is based on the methods presented by Erdman¹ and Crespo da Silva² in their respective books. The author believes that the employed approach is an excellent way to prepare students for future courses in multi-body dynamics and computational dynamics. It also makes students practice their knowledge of dynamics, numerical methods, and programming. Students' feedback regarding this project was overwhelmingly positive.

The project required students to design a, four bar, quick-return mechanism with a time ratio of 1.25 and a follower sweep angle of 50°. To check their work, students were asked to trace the paths of points on the crank, the coupler and the follower of their respective designs. A motor torque $M_{motor} = 1+2\sin(1.5t)$ N.m was then assumed to be applied to the crank, and students were asked to find the angular positions, velocities, and accelerations of each link using MATLAB and SIMULINK. Students also built their model in the shop.

Mechanism Synthesis

The complex number approach of Standard Dyad Form^{3,4,5} is used to perform the dimensional synthesis. This would render the size and starting configuration of the four bar function generation mechanism.

Figure 1 is a depiction of a typical function generation four bar mechanism. By applying the Loop-closure method to the vectors representing the links, in successive positions, a standard dyad form is resulted. Solution of the resulted vector equations then renders the size of each link of the mechanism and its starting configuration.

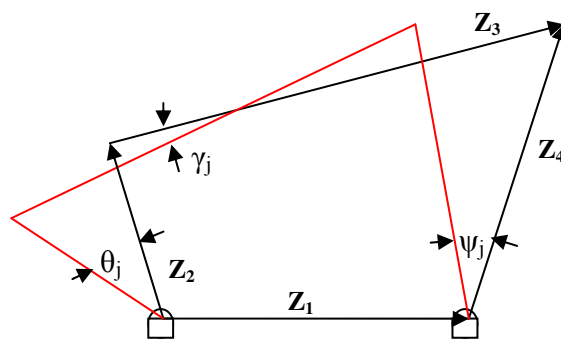


Figure 1

Figure 1 depicts the complex number vectors, Z_1, Z_2, Z_3, Z_4 , representing, the ground, the crank, the coupler and the follower links respectively, in the starting position. The red lines in the diagram are the depiction of the links in a different position as a result of the motion. The loop closure approach for position 1 (dark Vector lines) is:

$$Z_2 + Z_3 - Z_4 = Z_1 \quad (1)$$

The loop closure equation written again for the position 2 (red lines), in terms of the original position vectors, results in:

$$Z_2 e^{i\theta_j} + Z_3 e^{i\gamma_j} - Z_4 e^{i\psi_j} = Z_1 \quad (2)$$

Subtracting Equation (1) from Equation (2) renders:

$$Z_2 (e^{i\theta_j} - 1) + Z_3 (e^{i\gamma_j} - 1) = Z_4 (e^{i\psi_j} - 1) \quad (3)$$

Equation (3) represents a Standard Dyad Form. This equation is used to implement the design specifications described in this project.

Given the follower sweep angle of 50° , one realizes that, at the two extreme range of the follower motion, the mechanism is in toggle condition. That is the crank is lined up with the coupler. Since this is the only given in the problem, one has a two precision points description of the mechanism. This translates itself into adopting some free choices in order to solve equation (3) for the desired links' dimensions and orientations. Realizing that angle ψ_j for the follower in its sweeping motion is $\psi_j = 50^\circ$ and θ_j and γ_j angles are prescribed, once the applied time ratio (1.25) for the mechanism is implemented, one chooses vectors Z_3 and Z_4 freely to reduce the number of unknowns in equation (3) to one link only. The unknown (complex number) Z_2 is obtained from equation (3). If the rotation angle of the crank in the forward motion of the follower is taken as α and in the backward motion of the follower is taken as β we have:

$$\alpha + \beta = 360^\circ \quad (4)$$

The prescribed time ratio results in:

$$\frac{\alpha}{\beta} = 1.25 \quad (5)$$

Solution of (4) and (5) results in:

$$\alpha = 200^\circ \text{ and } \beta = 160^\circ \quad (6)$$

It can be easily shown that the angle, through which the coupler rotates when the mechanism moves from the first toggle condition to the next toggle condition is:

$$\gamma_j = \frac{\alpha - \beta}{2} = 20^\circ \quad (7)$$

This is achieved through crank rotation of:

$$\theta_j = \alpha = 200^\circ \quad (8)$$

With θ_j , γ_j , and ψ_j known as described in the above, one chooses vectors Z_4 and Z_3 in order to reduce the number of unknowns to one complex value for Z_2 in equation (3). Choosing:

$$Z_4 = 1e^{i65^\circ}, Z_3 = 1.43e^{i26.2^\circ} \quad (9)$$

Equation (3) after substitutions of the values renders:

$$Z_2(e^{i200^\circ} - 1) + 1.43e^{i26.2^\circ}(e^{i20^\circ} - 1) = 1e^{i65^\circ}(e^{i50^\circ} - 1) \quad (10)$$

Solution of the above equation results in:

$$Z_2 = 0.3463 \text{ m at } 26.2^\circ$$

With Z_2 , Z_3 , Z_4 , known, equation (1) is used to find the ground link Z_1 to be:

$$Z_1 = 1.178 \text{ m}$$

A CAD drawing of the mechanism is shown in Figure 2:

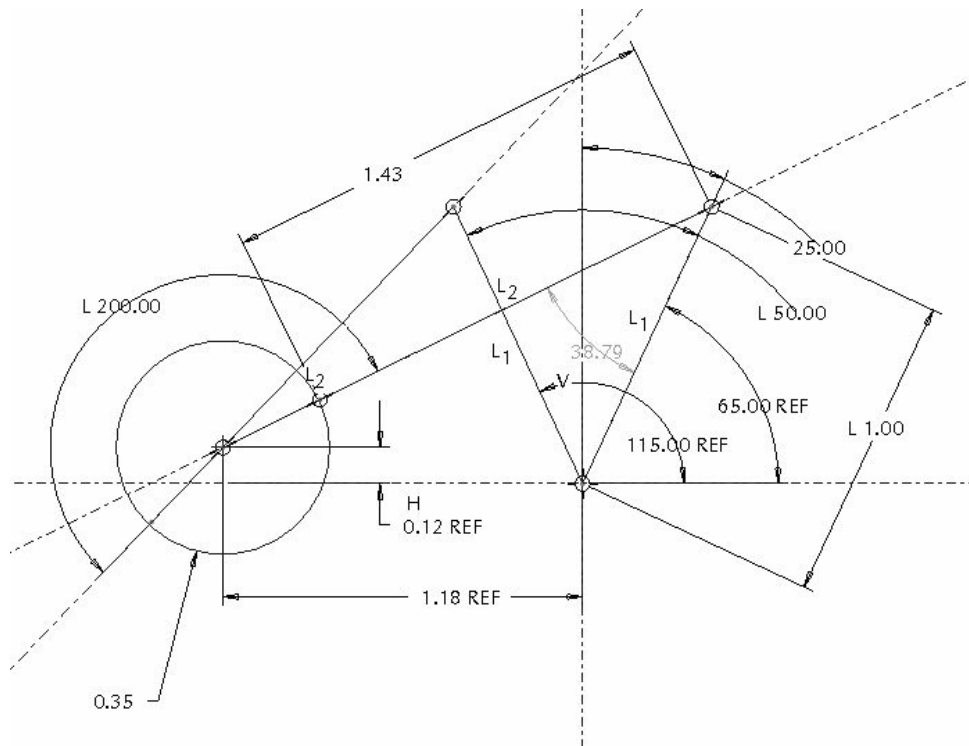


Figure 2

Once the mechanism was realized students were asked to find the position of each link and trace the crank curve, the curve for a point on the coupler, and the follower rocking

curve. This was done numerically, through the application of Newton-Raphson method⁶ to the scalar form of the loop-closure equation (1). The scalar form of equation (1) is:

$$\begin{aligned} r_2 \cos \theta_2 + r_3 \cos \theta_3 - r_4 \cos \theta_4 - r_1 &= 0 \\ r_2 \sin \theta_2 + r_3 \sin \theta_3 - r_4 \sin \theta_4 &= 0 \end{aligned} \quad (11)$$

Where r stands for length of each link and θ_2 , θ_3 , and θ_4 are the angles of links 2, 3, and 4 respectively.

Given θ_2 for the increments of motion of the crank shaft, nonlinear equations (11) are solved for the coupler and the follower angles. Appendix A includes the MATLAB program to fulfill this task. Figure 3 is the trace of the crank-tip, a coupler point (point C), and the follower tip of the mechanism. The program also confirms that the design meets the required specification and indicates that the follower sweeps a 50° angle in its ensuing motion.

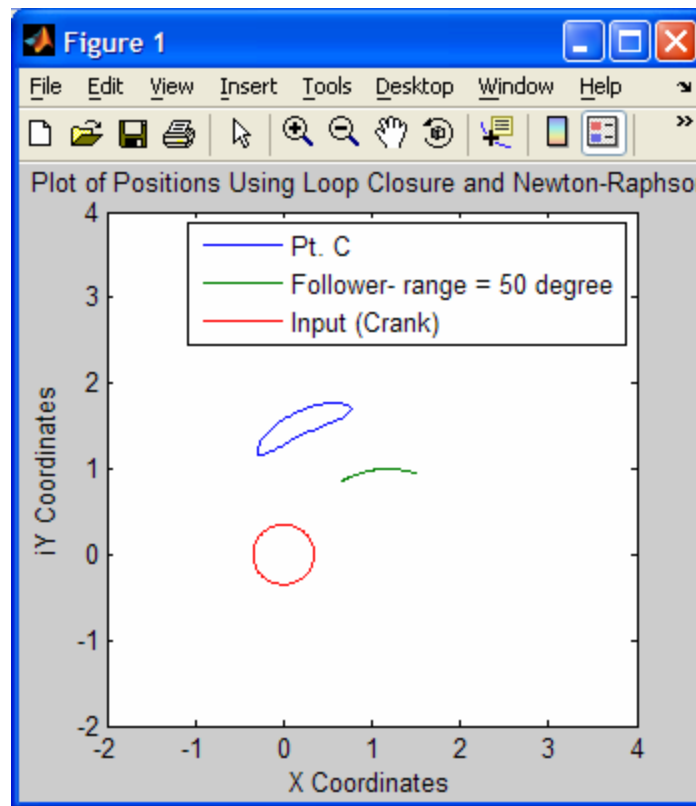


Figure 3

The next task was to find the angular velocities and accelerations of each link as part of the kinematics analysis of the mechanism. Differentiating equations (11) with respect to time renders equations for angular velocities of links 3 and 4, given the angular velocity of the link 2. These equations are:

$$\begin{aligned} -r_2\omega_2 \sin \theta_2 - r_3\omega_3 \sin \theta_3 + r_4\omega_4 \sin \theta_4 &= 0 \\ r_2\omega_2 \cos \theta_2 + r_3\omega_3 \cos \theta_3 - r_4\omega_4 \cos \theta_4 &= 0 \end{aligned} \quad (12)$$

Where $\omega_2, \omega_3, \omega_4$ are angular velocities of links 2, 3, and 4 respectively. Solving equations (12) for ω_3, ω_4 in terms of ω_2 one obtains:

$$\begin{bmatrix} -r_3 \sin \theta_3 & r_4 \sin \theta_4 \\ r_3 \cos \theta_3 & -r_4 \cos \theta_4 \end{bmatrix} \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = r_2 \omega_2 \begin{bmatrix} \sin \theta_2 \\ -\cos \theta_2 \end{bmatrix} \quad (13)$$

Equation (13) is solved for angular velocities of links 3 and 4, given the angular position of link 2 and its angular velocity ω_2 . This is accomplished in the MATLAB function of the SIMULINK model, presented later in the text.

To arrive at angular acceleration of the links when the motor torque is applied to the crank, one adopts Lagrange's approach to Figure (4) as follow:

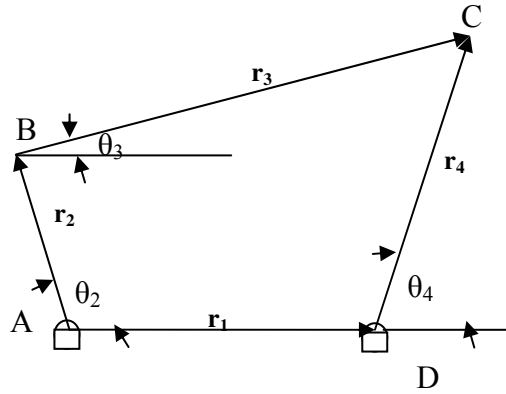


Figure 4

The Kinetic energy of the motion is obtained as:

$$T = \frac{1}{2} I_A \omega_2^2 + \frac{1}{2} I_3 \omega_3^2 + \frac{1}{2} I_D \omega_4^2 + \frac{1}{2} m_3 v_{cm3}^2 \quad (14)$$

Where I_A is the moment of inertia of link 2 about point A, I_D is the moment of inertia of link 4 about point D, and I_3 is the moment of inertia of link 3 about the center of mass of the link 3. V_{cm3} is the velocity of the center of mass for link 3. In the Cartesian coordinates V_{cm3} is:

$$v_{cm3} = \left(\frac{r_3}{2} \omega_3 \sin \theta_3 - \frac{r_4}{2} \omega_4 \sin \theta_4 \right) \vec{i} + \left(r_4 \omega_4 \cos \theta_4 - \frac{r_3}{2} \omega_3 \cos \theta_3 \right) \vec{j} \quad (15)$$

The virtual work on the system is simply:

$$\delta W = M_{motor} \delta \theta_2 \quad (16)$$

Where M_{motor} is the torque applied to the crank by the motor. The generalized coordinates θ_2 , θ_3 , and θ_4 are constrained by the loop closure equations (11). For the ease of following those equations are rewritten here:

$$\begin{aligned} f_1 &= r_2 \cos \theta_2 + r_3 \cos \theta_3 - r_4 \cos \theta_4 - r_1 = 0 \\ f_2 &= r_2 \sin \theta_2 + r_3 \sin \theta_3 - r_4 \sin \theta_4 = 0 \end{aligned} \quad (16)$$

Defining the Lagrangian as $L = T - V$, where T is the kinetic energy and V is the potential energy for the system. Neglecting the weight of the links in comparison with their respective inertia forces, the potential energy of the system is set to zero. With that in mind, the Lagrange's equation for each generalized coordinate is then written as follows:

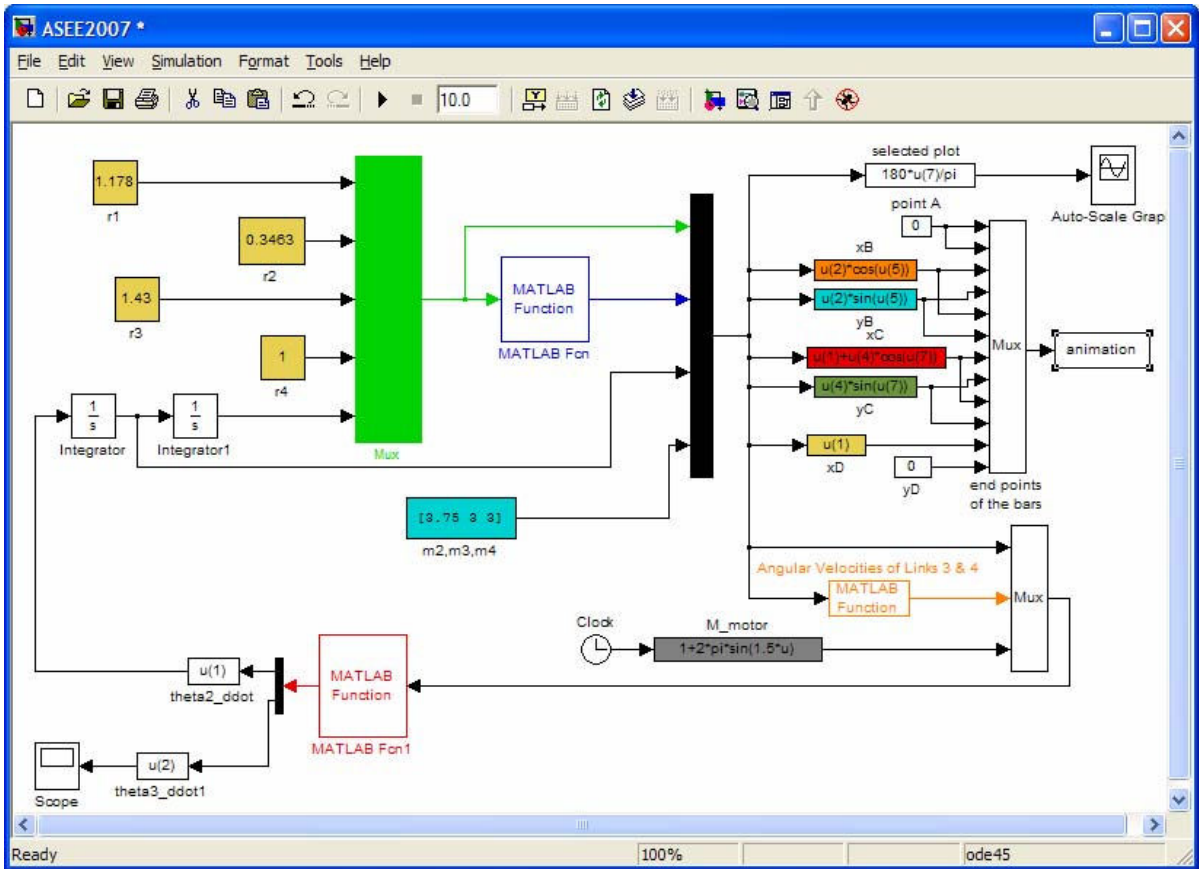
$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}_2} \right) - \frac{\partial T}{\partial \theta_2} &= M_{motor} + \lambda_1 \frac{\partial f_1}{\partial \theta_2} + \lambda_2 \frac{\partial f_2}{\partial \theta_2} \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}_3} \right) - \frac{\partial T}{\partial \theta_3} &= \lambda_1 \frac{\partial f_1}{\partial \theta_3} + \lambda_2 \frac{\partial f_2}{\partial \theta_3} \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}_4} \right) - \frac{\partial T}{\partial \theta_4} &= \lambda_1 \frac{\partial f_1}{\partial \theta_4} + \lambda_2 \frac{\partial f_2}{\partial \theta_4} \end{aligned} \quad (17)$$

Where λ_1 and λ_2 are Lagrange multipliers, due to the constraint motion. Substituting the links' moments of inertia in terms of their lengths and masses ($I_A = (1/3)m_2r_2^2$, $I_D = (1/3)m_4r_4^2$, $I_3 = (1/12)m_3r_3^2$) into the expression for the kinetic energy, performing the partial derivatives of the constraint equations (16) with respect to θ_2 , θ_3 , and θ_4 , and plugging the results into equations (17); one arrives at the expressions for the angular accelerations of links 2, 3, and 4, namely $\ddot{\theta}_2$, $\ddot{\theta}_3$, and $\ddot{\theta}_4$. The outcome of these mathematical operations would be 3 equations of (17) in terms of the 5 unknowns, $\ddot{\theta}_2$, $\ddot{\theta}_3$, $\ddot{\theta}_4$, λ_1 , and λ_2 .

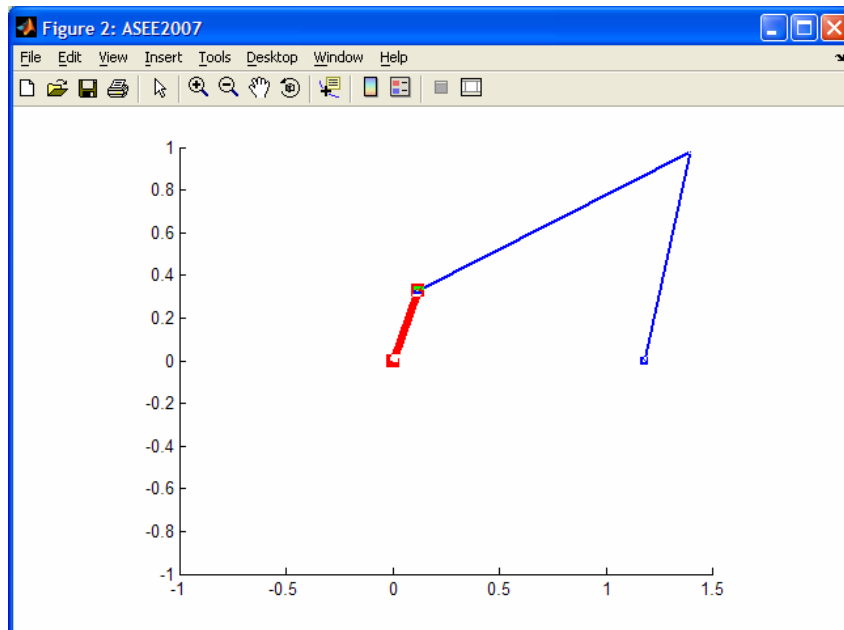
In order to solve for the angular accelerations of the links 2, 3, and 4, one needs to supplement the reduced form of the equations (17) with two more equations. This is done by differentiating the equations of constraints, equations (16), twice with respect to time. The result of which is:

$$\begin{aligned} r_2 \ddot{\theta}_2 \sin \theta_2 + r_3 \ddot{\theta}_3 \sin \theta_3 - r_4 \ddot{\theta}_4 \sin \theta_4 + r_2 \dot{\theta}_2 \cos \theta_2 + r_3 \dot{\theta}_3 \cos \theta_3 - r_4 \dot{\theta}_4 \cos \theta_4 &= 0 \\ r_2 \ddot{\theta}_2 \cos \theta_2 + r_3 \ddot{\theta}_3 \cos \theta_3 - r_4 \ddot{\theta}_4 \cos \theta_4 - r_2 \dot{\theta}_2 \sin \theta_2 - r_3 \dot{\theta}_3 \sin \theta_3 + r_4 \dot{\theta}_4 \sin \theta_4 &= 0 \end{aligned} \quad (18)$$

One should note that equations (18) are nonlinear and carry the angular positions and velocities of the links 2, 3, and 4 in them. This means that the angular positions and velocities of the links 3, and 4 need to be known prior to the solution for the angular accelerations of all the links. Solution of these equations (reduced form of equations (17) and (18)) would be implemented by SIMULINK as shown below.



The MATLAB functions with foreground colors, blue, orange and red, seen in the SIMULINK model, are MATLAB .m files for finding positions, velocities and accelerations of the links respectively. The program also animates the mechanism. A snapshot of such an animation is shown below:



Appendix B contains the three MATLAB Functions used in the SIMULINK model. Figure 5 is the Auto-Scale Graph of the SIMULINK model, which is the plot of the angular position of link 4 (θ_4) vs. time. Figure 5 confirms the rocking motion of the follower. It also shows that the follower sweeps an angle of 50° in its rocking motion.

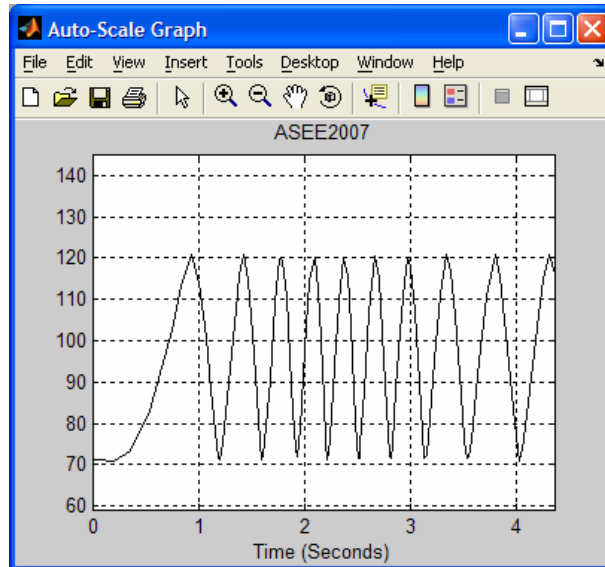


Figure 5

Conclusion

The project significantly helped students understand the abstract concepts in dynamics. This was reflected in the result of the follow up exam. Majority of the students exhibited a very thorough understanding of Lagrange's equations. Students enjoyed the animation part of the project and built their models in the shop. The author received positive feedback from the students regarding this exercise.

Bibliography

1. Erdman, Arthur G., Sanders, George N., "Mechanism Design Analysis and Synthesis" Prentice Hall, 2001.
2. Crespo da Silva, M. R.M., "Intermediate Dynamics", McGraw Hills, 2004.
3. Erdman, Arthur G, "Three and Four Precision Point Kinematics Synthesis of Planar Linkages", Mechanism and Machine Theory, 16, pp. 227-45.
4. Erdman, Arthur G, "Three Position Synthesis by Complex Numbers", In Monograph on Mechanical Design, Paper No. 49, 1977.
5. Erdman, Arthur G, and Carlson, W. L., "Teaching Unit on Complex Numbers as Applied to Linkage Modeling", In Monograph on Mechanical Design, NSF report No. GK36624, Paper No. 12, 1977.
6. Moore, Holly "MATLAB for Engineers", Prentice Hall, 2007.

Appendix A

```
%*****
%
%
%   Path Generation Program Using Loop Closure Method
%
%
%*****
% FILENAME: path_loop_closure.m

% CREATE MATRICES TO STORE X AND iY COMPONENTS OF POSITION OF PTS. A,
B,
% AND C AND THETA 3 AND 4 ESTIMATES
Xa=[];
Ya=[];
Xb=[];
Yb=[];
Xc=[];
Yc=[];
thetabars=[];

% DEFINE CONSTANTS (LENGTHS IN INCHES, ANGLES IN RADIANS)
r1=1.178;           % "G" Ground Link Length, AoBo
r2=0.3463;         % "U" Input Link Length, AoA
r3=1.43;           % "Z" Coupler Link Length, AB
r4=1;               % "W" Follower Link Length, BoB
r5=1.54;           % Length AC
theta1=0;          % Angle of Ground Link
psic=40.6*(pi/180); % Angle BAC
mu=-1;             % Configuration of linkage

% Grashof (s+l<p+q since r2+r1<r3+r4) and input is the shortest
% link => Crank Rocker (Crank can rotate 360 degrees)
theta2min=0;       % Smallest input angle
theta2max=2*pi;    % Largest input angle
range=theta2max-theta2min; % Range of input motion
steps=100;         % Number of positions that will be
calculated

% CALCULATE INITIAL POSITION OF C WITH COMPLEX NUMBERS
theta2=theta2min;  % Initial theta2
r2v=r2*exp(i*theta2); % Position vector AoA
r1v=r1*exp(i*theta1); % Position vector AoBo
r7v=r2v-r1v;       % Position vector BoA
r7=abs(r7v);       % Magnitude BoA
psi=acos((r4^2+r7^2-r3^2)/(2*r4*r7)); % Angle ABoB
theta4=imag(log(r7v/abs(r7v)))+mu*psi; % Current theta4
r4v=r4*exp(i*theta4); % Position vector BoB
r3v=r1v-r2v+r4v;   % Position vector AB
theta3=imag(log(r3v/abs(r3v))); % Angle of AoA to X axis

% CALCULATE POSITION OF C AT ALL STEPS
for q=1:(steps+1)
```

```

theta2=theta2min+(q-1)*(range)/steps;    % Current theta2

% CALL FUNCTION TO GET ESTIMATES OF THETAS 3 AND 4
thetabars=thetas(theta1,theta2,theta3,theta4,r1,r2,r3,r4);
theta3=thetabars(1);    % Set current theta3 to Newton-Raphson
estimate
theta4=thetabars(2);    % Set current theta4 to Newton-Raphson
estimate
thth(q)=theta4;

Xc(q)=r2*cos(theta2)+r5*cos(theta3+psic); % Put current Xc in
matrix
Yc(q)=r2*sin(theta2)+r5*sin(theta3+psic); % Put current iYc in
matrix
Xb(q)=r1*cos(theta1)+r4*cos(theta4);     % Put current Xb in
matrix
Yb(q)=r1*sin(theta1)+r4*sin(theta4);     % Put current iYb in
matrix
Xa(q)=r2*cos(theta2);                    % Put current Xa in
matrix
Ya(q)=r2*sin(theta2);                    % Put current iYa in
matrix
end
theta4max=max(thth);
theta4min=min(thth);
rangel=(theta4max-theta4min)*180/pi

% PLOT THE POSITIONS OF C, B, AND A
plot(Xc,Yc,Xb,Yb,Xa,Ya);
title('Plot of Positions Using Loop Closure and Newton-Raphson');
axis([-2,4,-2,4]);
xlabel('X Coordinates');
ylabel('iY Coordinates');
legend('Pt. C','Follower- range = 50 degree','Input (Crank)');
animate_nbar

```

```

%*****
%
%
%
% Function for Path Generation Program Using Loop Closure Method
% 2-12-06
%
%*****
% FILENAME: thetas.m

% FUNCTION FINDS NEWTON-RAPHSON APPROXIMATION OF THETAS 3 AND 4
% BASED ON PREVIOUS ANGLES AND BASED ON LINK MAGNITUDES
function y=thetas(th1,th2,th3,th4,m1,m2,m3,m4)

% SET ESTIMATES EQUAL TO LAST THETAS 3 AND 4
theta3bar=th3;
theta4bar=th4;
%INITIALIZE MATRIX TO STORE X AND Y SUMS
F=[1;1];

% LOOP UNTIL MAGNITUDE OF X AND Y SUMS IS VERY SMALL -- NEAR ZERO
while norm(F)>=1.0e-010 %if eps, program looped forever

% X COMPONENTS AT CURRENT ESTIMATE (MUST ADD UP TO ZERO)
f1=m2*cos(th2)+m3*cos(theta3bar)-m4*cos(theta4bar)-m1*cos(th1);
% Y COMPONENTS AT CURRENT ESTIMATE (MUST ADD UP TO ZERO)
f2=m2*sin(th2)+m3*sin(theta3bar)-m4*sin(theta4bar)-m1*sin(th1);
% JACOBIAN DETERMINATE IS CALCULATED
A=[(-m3*sin(theta3bar)) (m4*sin(theta4bar));(m3*cos(theta3bar))
(-m4*cos(theta4bar))];
%THE X AND Y AT CURRENT ESTIMATE
b=[(-(f1));(-(f2))];
% MATRIX "DIVISION" -- EQUIVALENT TO A^-1*b, BUT FASTER
EXECUTION
x=A\b;
% NEW ESTIMATE OF THETAS 3 AND 4
theta3bar=theta3bar+x(1,1);
theta4bar=theta4bar+x(2,1);
% NEW SUM OF X AND Y COMPONENTS
f1=m2*cos(th2)+m3*cos(theta3bar)-m4*cos(theta4bar)-m1*cos(th1);
f2=m2*sin(th2)+m3*sin(theta3bar)-m4*sin(theta4bar)-m1*sin(th1);
% PUT X AND Y SUMS IN MATRIX SO NORM CAN BE COMPUTED
F=[f1;f2];

end

% FINAL ESTIMATES ARE RETURNED AS A VECTOR
y=[theta3bar theta4bar];

end

```

Appendix B

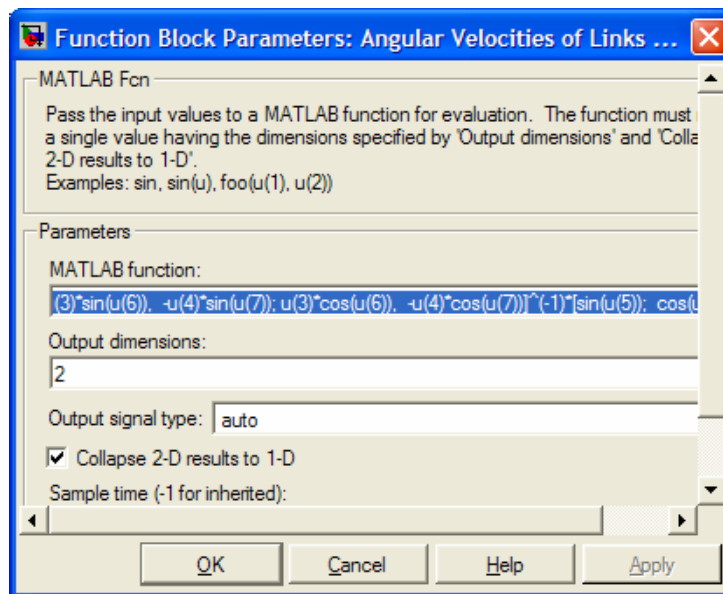
The Blue MATLAB Function in the SIMULINK Model

```
% This Function is the blue MATLAB Function in the SIMULINK model to  
% calculate the angular position of the links. It is based on the  
% Newton-Raphson method for solving system of nonlinear equations
```

```
function [th]=Newton(u)  
factor=pi/180;  
theta2=u(5);  
theta3=45*factor;  
theta4=135*factor;  
f=[u(3)*cos(theta3)-u(4)*cos(theta4)+u(2)*cos(theta2)-u(1);  
u(3)*sin(theta3)-u(4)*sin(theta4)+u(2)*sin(theta2)];  
while norm(f)>1.0E-6  
    Jacobian=[-u(3)*sin(theta3) u(4)*sin(theta4); u(3)*cos(theta3) -  
u(4)*cos(theta4)];  
    delta= inv(Jacobian)*-1*f;  
    theta3=theta3+delta(1);  
    theta4=theta4+delta(2);  
    f=[u(3)*cos(theta3)-u(4)*cos(theta4)+u(2)*cos(theta2)-u(1);  
u(3)*sin(theta3)-u(4)*sin(theta4)+u(2)*sin(theta2)];  
norm(f);  
end;  
th(1)=theta3;  
th(2)=theta4;
```

The Orange MATLAB Function in the SIMULINK Model

Given the Angular Velocity of the Link 2, this Function Calculates the Angular Velocities of the Other Links



The Red MATLAB Function in the SIMULINK Model

```
% This Function is the red MATLAB Function in the SIMULINK model, used
% to find the angular accelerations of the links

function [acc] = accel(u)
A = [1/3*u(9)*u(2)^2 0 0 u(2)*sin(u(5)) -u(2)*cos(u(5));0
1/3*u(10)*u(3)^2 -1/2*u(10)*u(3)*u(4)*cos(u(7)-u(6)) u(3)*sin(u(6)) -
u(3)*cos(u(6));
0 -1/2*u(10)*u(3)*u(4)*cos(u(7)-u(6)) (1/3*u(11)+u(10))*u(4)^2 -
u(4)*sin(u(7)) u(4)*cos(u(7));
u(2)*sin(u(5)) u(3)*sin(u(6)) -u(4)*sin(u(7)) 0 0; u(2)*cos(u(5))
u(3)*cos(u(6)) -u(4)*cos(u(7)) 0 0]

B = [u(14); -1/2*u(10)*u(3)*u(4)*u(13)^2*sin(u(7)-u(6));
1/2*u(10)*u(3)*u(4)*u(12)^2*sin(u(7)-u(6));
u(4)*u(13)^2*cos(u(7))-u(3)*u(12)^2*cos(u(6))-
u(2)*u(8)^2*cos(u(5));
-
u(4)*u(13)^2*sin(u(7))+u(3)*u(12)^2*sin(u(6))+u(2)*u(8)^2*sin(u(5))]
acc = (A^-1)*B
```