DFA Lab - Background Information

Dynamic force analysis is a technique of calculating node forces and torques throughout a mechanism (summing x & y forces and moments) to fulfill the observed kinematics (both linear and angular). In doing so, we can calculate the relative forces on the joints, or nodes, throughout the mechanism which can give insight on possible failure locations and predict input torques needed to drive the mechanism. More specifically, the procedure discussed here solves the inverse dynamics problem which takes the dynamics information from the PVA code and information of the link's masses and moments of inertias to solve for internal forces and torques.

A completed version of the PVA code for this lab exercise has been provided to you. The mechanism is a simple four-bar mechanism where link 2 is driven by a constant velocity motor. Hence, there will be a torque at that link.

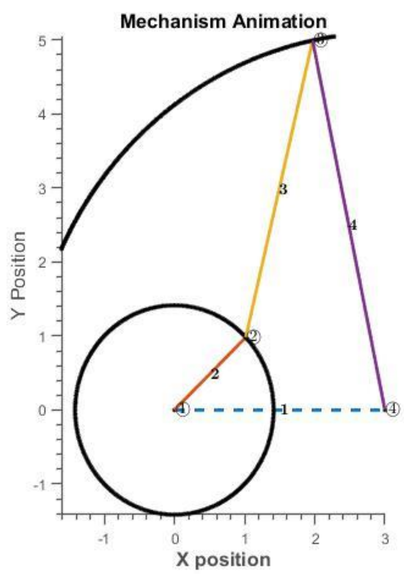

Figure 1: Linkage used in this lab exercise

Figure 2 provides an exploded view of the mechanism with internal joint forces according to our labeling scheme. F_{ij} should be read as the force that link i exerts on link j. Doing this for each link as shown gives 8 unknown force variables with 2 components each (x and y) resulting in 16 unknown force components. The unknown input motor torque needed to maintain the constant motor angular velocity assumed in the original PVA code must be included as well. This gives 17 unknown variables. However, summing the forces at the nodes will eliminate 8 of these unknowns (for instance, $F_{12x} = -F_{21x}$ and $F_{12y} = -F_{21y}$). This gives us a total of 9 unique unknowns for this problem.

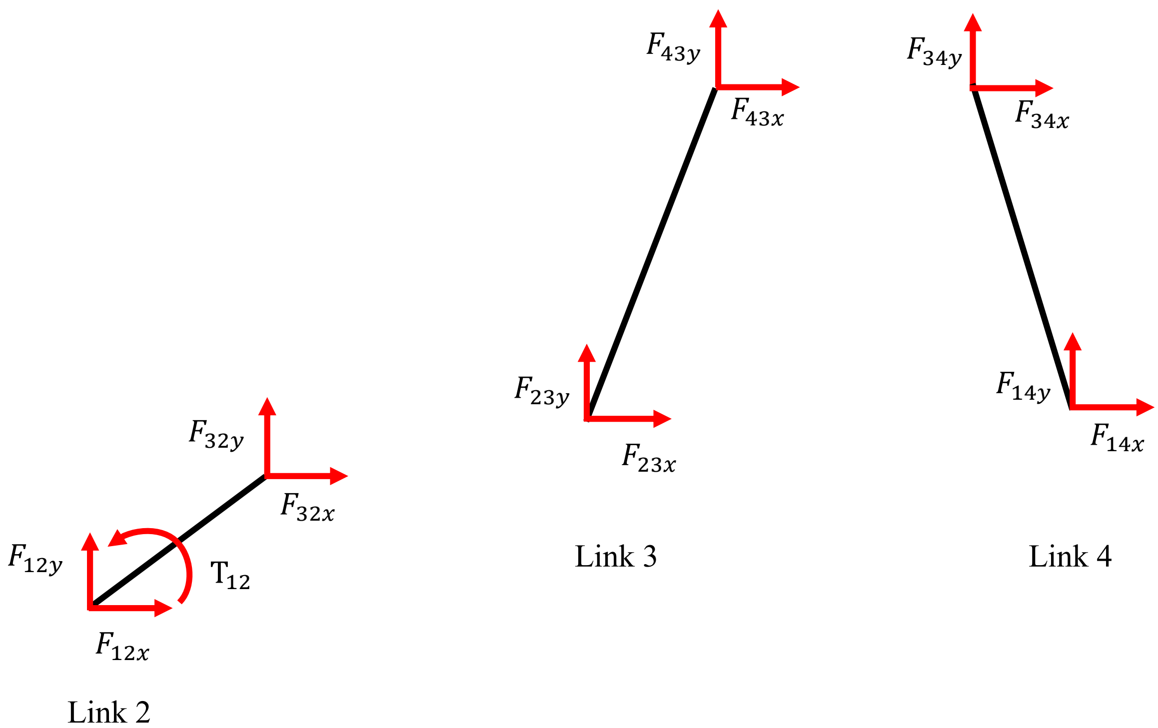

Figure 2: Links 2-4 with internal joint forces and applied crank torque

There are three moving links, each of which has three associated DFA equations: ΣF_x, ΣF_y and ΣT. This gives us 9 equations. Hence, the solution can be obtained.

The procedure behind the dynamic force analysis is to take all the known equations describing the force balance on the system and summarize them in matrix notation for solving. The equation [A]{B}={C} shows the form of this process where [A] is a matrix and {B} and {C} are single column arrays as shown in the matrix below. B is a vector holding the unknown force or torque components that we want to solve for. Each line of this expression summarizes a different equation or relationship in the system. For example, the first three dynamic equations for link 2 are given as:

$$\Sigma F_x \Rightarrow F_{12x} + F_{32x} = m_2 a_{CG2,x} \tag{1}$$
$$\Sigma F_y \Rightarrow F_{12y} + F_{32y} = m_2 a_{CG2,y} + m_2 g \tag{2}$$
$$\Sigma T \Rightarrow T_{12} + R_{12x}F_{12y} - R_{12y}F_{12x} + R_{32x}F_{32y} - R_{32y}F_{32x} = I_{CG2}\alpha_2 \tag{3}$$

where g is the gravitational constant (+9.81m/s^2), R_{ij} is the vector from the center of gravity of link j to the node where F_{ij} acts, a_{CG2} is the linear acceleration of the center of gravity of Link 2, I_{CG2} is the moment of inertia of link 2, α_2 is the angular acceleration of link 2. **For this exercise, we are assuming that the links have weight, i.e. potential energy is not neglected.** The weights act downwards at the center of mass of the links. The DFA code sums the forces in x & y and the torques for each link in order of link number. Notice the equations corresponding to the ground link are not included in the matrix because it is fixed and not moving. This gives 9 equations as mentioned before.

The system, therefore, has 9 equations and 9 unknowns and can be solved. In Python, the matrix-vector equation [A]{B}={C} can be solved using {B}=[A]$^{-1}${C}. The process of constructing and solving the matrix-vector relations is performed at each time-step of the simulation and stored in the array B, where each column represents the solution vector at a different time-step.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -R_{12y} & R_{12x} & -R_{32y} & R_{32x} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & R_{23y} & -R_{23x} & -R_{43y} & R_{43x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & R_{34y} & -R_{34x} & -R_{14y} & R_{14x} & 0 \end{bmatrix} \begin{Bmatrix} F_{12x} \\ F_{12y} \\ F_{32x} \\ F_{32y} \\ F_{43x} \\ F_{43y} \\ F_{14x} \\ F_{14y} \\ \mathrm{T}_{12} \end{Bmatrix} = \begin{Bmatrix} m_2 a_{CG2,x} \\ m_2 a_{CG2,y} + m_2 g \\ I_{CG2}\alpha_2 \\ m_3 a_{CG3,x} \\ m_3 a_{CG3,y} + m_3 g \\ I_{CG3}\alpha_3 \\ m_4 a_{CG4,x} \\ m_4 a_{CG2,y} + m_4 g \\ I_{CG4}\alpha_4 \end{Bmatrix}$$

As an example, the entries for all equations are included in the above matrix. If you multiplied out the matrix-vector expression you can easily confirm this. The solution vector in this process is ordered as seen above where the forces at each link’s node are ordered (x then y component). The motor torque is then included at the end.

The code is explained as follows:

Section 0: Imports Libraries

The position, velocity, and accelerations of each link (and their nodes) are computed for all time-steps of the simulation from the PVA code.

Section 1: Input

cmlink is a single column array that holds a value for each link (0-1) which represents the fractional link distance from the first node entry that the link's center of mass is. For example, assuming a row entry in cmat, [4 3], which states that the link is made between the connection of nodes 4 and 3, a corresponding *cmlink* entry of 0.5 would put the center of mass for the link halfway between the nodes 4 and 3. Furthermore, an entry of 0 would imply the center of mass is on node 4 and an entry of 1 would put the center of mass on node 3.

cmatbody is an array that defines each rigid body. In the PVA code, only binary links are defined between two nodes. A fixed angle constraint is used for combining such links to physically behave as one link. The *cmatbody* array is used for this purpose.

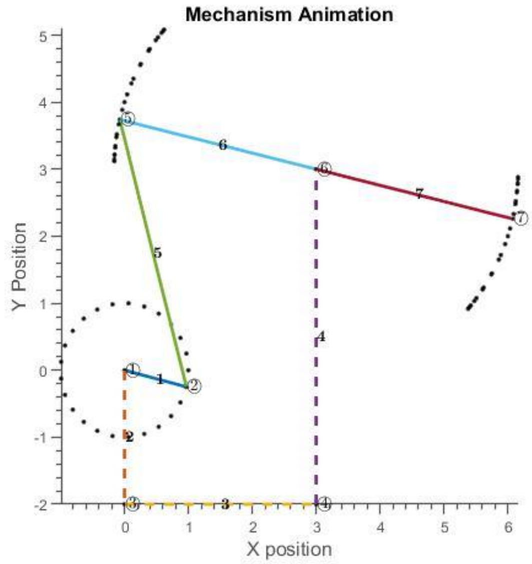

Figure 3: Illustration of fixed angle constraint between links 6 and 7

Figure 3 is an example where this would be used. Here, links 6 and 7 are physically the same link. So, *cmatbody* would be [1, 2, 3, 4, 5, [6, 7]], giving you 6 bodies. Here each column represents the "links" from the PVA that combine to behave as one. **Define each ground link individually (if there is more than one).**

gconst: gravitational constant in m/s^2 (positive value)
wdth: width of each link in m.
tks: thickness of each link in m.
den: density for each link in kg/m^3.

The rest of the code is structured as follows. Find the section headers in the code and read the accompanying code comments to better understand what each line is doing.

Section 2: Calculates the mass of each link/body
Mlink: Calculates the mass of each link as defined in the PVA. For a cuboid, it is simply $den * wdth * tks * length$. The length of each link can be obtained from the links vector in the PVA (Recall Prelab 6).

M: Calculates the mass of each body as defined in *cmatbody.*

Section 3: Calculates the center of mass and moment of inertia for each link/body
The position of the center of mass for each body is calculated in Section 3. Using this, the moments of inertia are also calculated. The parallel axis theorem is used if modelled by multiple links.

Section 4: Calculates $\theta, \omega\ and\ \alpha$ of each link

Since the coordinates of all nodes are known, θ for each link can be easily determined. $\omega\ and\ \alpha$ are calculated by differentiating with respect to time (similar to velocity and acceleration in the PVA code).

Note: For a body that consist of several "links", the choice of θ does not matter, since $\omega\ and\ \alpha$ for a rigid body is the same at every location.

Section 5: Calculates the velocity and acceleration of each body's center of mass

The position of each center of mass was obtained in Section 3. This is also differentiated with respect to time to get the corresponding velocity and acceleration (similar to the PVA code).

Section 6: Initializes the B vector, external forces and moving bodies

There are no external forces acting on the mechanism in question. If there are, these are defined as complex numbers in a column array *F*. The body (according to *cmatbody*) and the node (according to *xnode*) where each external force acts is also defined here as column arrays. If an external force acts on the center of mass of a body, leave that node value as zero.

Section 7: Solves the equation [A]{B} = {C}

Section 8: Checks for power conservation using the Principle of Virtual Work

Calculate the power due to the external forces and torques ($\Sigma(\vec{F} \cdot \vec{v}) + \Sigma(\vec{T} \cdot \vec{\omega})$) and check if it overlaps with the power from the inertial properties of the links ($\frac{\mathrm{dE}}{\mathrm{dt}}$), where E is the total energy of the system ($KE_{rot} + KE_{trans} + PE$).

***Note*:** In lecture, the potential energy is neglected in the virtual work formulation. However, for this problem, we have assumed the links to have a non-zero weight. **So, $\boldsymbol{\Sigma(\mathbf{m}\overrightarrow{\mathbf{A_{CG}}} \cdot \overrightarrow{v_{CG}}) + \Sigma(\mathbf{I_{CG}}\vec{\alpha} \cdot \vec{\omega})}$ will not be an accurate representation of $\boldsymbol{\frac{\mathrm{dE}}{\mathrm{dt}}}$ for this lab exercise.**

Section 9: Generates the required plots.

DFA Lab – Pre-Lab Exercise

Read this manual and complete the Pre-lab assignment posted on Gradescope before the respective due dates. All the answers can be found in this lab document.

DFA - In-lab Exercise (30 points)

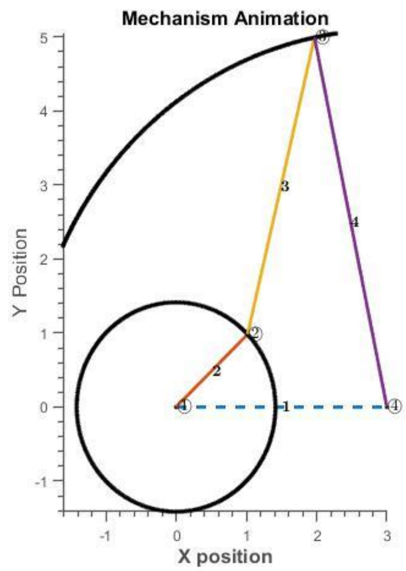

Figure 1: Linkage used in this lab exercise

Before starting:

1. *All code templates are available at the link:*
https://drive.google.com/file/d/1vBjPv6w9eT80GrwOVZQgSG1zKfTnJMbd/view?usp=sharing
2. *Make a local copy of the ME370_DFA_in_lab.ipynb notebook to your Google colaboratory. To make a local copy, download the notebook from the link above, open Google Colaboratory in your browser, go to File->Upload notebook (you may need to sign-in).* ***You should not modify the first 3 cells.***
3. *Note: This is an* ***individual*** *lab, and you need to fill out the gradescope questions individually.*

Question 1 (6 points)

a) Run the code cell below Section 0 without modification.
b) Complete **Section 1** of the DFA code. Assume that each link has a center of mass halfway between the nodes. Add a center of mass location entry for the ground link as well, though the value for that link doesn't matter. The width, thickness and density of the links are 1 inch (0.0254m), 1/8 inch (0.003175m) and 1180 kg/m^3 (acrylic) respectively. Comment your code thoroughly. In the comments, explain how your code works.
c) Include answer in post-lab report (Screenshot of changed code).
d) Run this code cell.

Question 2 (6 points total)

a) Run the code cell below Section 2 without modification.

b) Complete **Section 3** of the DFA code, which computes the position of the center of mass and the moments of inertia for each body (3 points). Comment your code thoroughly. In the comments, explain how your code works.
c) Run this code cell.
d) Complete **Section 4** of the DFA code, which computes the angular position, velocity, and acceleration (3 points). Comment your code thoroughly. In the comments, explain how your code works.
e) Run this code cell.
f) Include answer in post-lab report (Screenshot of changed code).

Question 3 (6 points)

a) Run the code cell below Section 5 without modification.
b) Run the code cell below Section 6 without modification.
c) Complete **Section 7** of the DFA code, which defines position vectors for the torque equations and the A & C arrays. Comment your code thoroughly. In the comments, explain how your code works.
d) Run this code cell.
e) Include answer in post-lab report (Screenshot of changed code).

Question 4 (6 points)

a) Complete **Section 8** of the DFA code. Comment your code thoroughly. In the comments, explain how your code works.
b) Run this code cell.
c) Include answer in post-lab report (Screenshot of changed code and output).

Question 5 (6 points)

a) **Plot the input torque required to run the motor versus crank angle**. To do so, you'll need to identify where in the solution vector the motor torque component appears.
b) You have been provided with a few lines of code to **compare plots of input power, T_{12}*(ω_{crank}), to the time rate of change of the energy of the entire mechanism**. Run this code cell to observe that the system conserves power.
c) Include answer in post-lab report (Screenshot of changed code and the three plots generated here). **There is no need to include the PVA plots or animation.**

Select one of the following options:
a) My answer was created by a Gen AI algorithm, and I have not modified it
b) My answer was created by a Gen AI algorithm, and I have made some minor changes.
c) My answer was created by a Gen AI algorithm, and I have made major changes.
d) My answer was created solely by myself.
e) If I used Gen AI, I used ___ (name of program).

Lab: Dynamic Force Analysis

DFA Post-lab Assignment (30 pts)

The force F1 acts at point P as shown (Figure 1). F1 has a magnitude of 80 N, and the angle it makes with the horizontal is a constant 45°. You may assume ω_2 to be 30 rpm (constant) and θ_{O2O4} to be 45°. All lengths are in meters. Use the PVA code (PVA_ChebyshevCircle.py) provided to you on canvas to obtain the kinematic characteristics of the mechanism and generate the matrices required for the DFA calculation. Then, use the DFA code to solve for the forces, neglecting gravity. You may assume links 3, 4, and the ground to be uniform and cuboidal, with a width and thickness of 0.1 m. The crank is cylindrical with a thickness of 0.1 m into the plane. Assume that all links are made of aluminum, which has a density of 2500 kg/m^3.

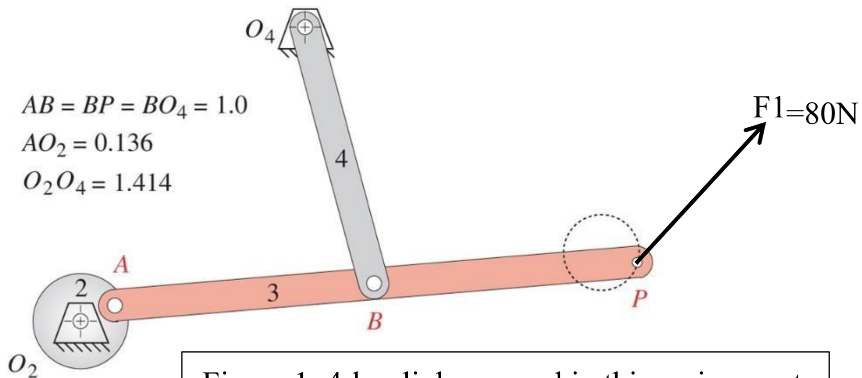

Figure 1: 4-bar linkage used in this assignment.

Before starting:

1. *All code templates are available at the link:*
https://drive.google.com/file/d/16-iDA4-_DdLalX3J8bdygoNQq6VH8zaA/view?usp=sharing

2. *Make a local copy of the ME370_DFA_postlab.ipynb notebook to your Google colaboratory. To make a local copy, download the notebook from the link above, open Google Colaboratory in your browser, go to File->Upload notebook (you may need to sign-in).* ***You should not modify the first 3 cells.***
3. *Note: This is an* ***individual*** *lab, and you need to fill out the gradescope questions individually.*

Question: Repeat the same questions 1-5 as you answered for your in-lab activity for this post-lab activity as well.

Deliverables: In addition to the in-lab code screenshots and plots, include the following from this postlab assignment:

1. Plot #1 (5 points): Plot of the motor torque at link 2 as a function of the time for 3 full cycles. Be sure to include **axes labels** with **specified units**!
2. Plot #2 (5 points): Plot of a comparison of input power (from external torques and forces) to the time rate of change of energy of the entire mechanism with time on the x-axis. Include **axes labels**

with **specified units**! __**Note:**__ Your plots will be similar but not identical to each other due to the input power having an F•v term.

3. Plot #3 (5 points): Plot of the input torque required to run the motor versus crank angle. Be sure to include **axes labels** with **specified units**!
4. The postlab code (only the modified one) at the end of your PDF submission. **Comment** on the modifications and additions you had to make to the code in order to model this linkage. (15 points) **There is no need to include the PVA plots or animation.**

Select one of the following options:
a) My answer was created by a Gen AI algorithm, and I have not modified it
b) My answer was created by a Gen AI algorithm, and I have made some minor changes.
c) My answer was created by a Gen AI algorithm, and I have made major changes.
d) My answer was created solely by myself.
e) If I used Gen AI, I used ___ (name of program).

__Tips:__

- Neglecting gravity does not mean that the links have no mass. It means that potential energy is negligible (hence it does not appear in the virtual work formulation), so g is zero.
- The moment of inertia for a cylinder about its axis of rotation = $0.5mr^2$, where r is the radius of the cylinder. For a cuboid about its center, $I_C = m(w^2 + l^2)/12$.
- $\sum(m\vec{A}_{CG} \cdot \overrightarrow{v_{CG}}) + \sum(I_{CG}\vec{\alpha}_{CG} \cdot \vec{\omega}) = \frac{dKE}{dt}$, where KE is the sum of the translational and rotational kinetic energies. It may be easier to solve for $\frac{dKE}{dt}$ using the given code.
- Don't forget that the center of mass of link 2 is at O_2 for all time.