Mechanisms and Machines ENSC3001

Second Semester Deadline:

Monday, October 4th, 2021

Coupler curves and design of a linear pumping mechanism

(20% of the total mark)

The aim of this project is to practice analytical methods of kinematic analysis of mechanisms, as well as reinforce computer programming skills.

The project will be conducted in groups of five (groups of four are acceptable, groups of six are NOT). One submission per group is required. Individual contributions of group members to the project will be assessed using the self and peer assessment tool Spark^{PLUS}.

A short video of the desired mechanism can be found on LMS.

Problem

Coupler curves and design of a linear pumping mechanism

I. Write a program to draw coupler curves for 4-bar mechanisms (any programming language is acceptable; we suggest MATLAB if you don't have prior programming experience). It should contain the following components:

1. Input dimensions of the linkage, assembly mode and position of a coupler point

2. Find motion limits (Theta2min, Theta2max)

3. Compute the position of the coupler point for Theta2 running from Theta2min to Theta2max and plot the result

Remember to document your code carefully, so that other people can use (or check) it!

Additional programming resources can be found here:

Online MATLAB training: https://matlabacademy.mathworks.com/details/matlab-onramp/gettingstarted

Good primer on MATLAB programming: http://www.mathworks.com/moler/chapters.html

II. Choose dimensions for the four-bar linkage of the linear pumping mechanism (you may use a 4-bar software to help you in the design). Your linkage should produce a coupler curve, which contains a substantial vertical straight line. Using your program show that the coupler point, from which the load is suspended, indeed moves approximately along a vertical straight line.

III. **Your program should be able to deal with all four-bar linkage types.** It should also give appropriate warning messages when the user input data is incorrect – i.e. somewhere within the range of theta2min and theta2max the dimensions of the mechanism are inconsistent with the geometry required by theta2. You will have seen in Geogebra that for some combinations of dimensions the links, the crank-link cannot rotate a full 360 degrees without the mechanism breaking.

IV. It is very important that your program is well documented so that other people (e.g. your marker) can run or check it.

Milestones:

Milestone One - deadline Monday, September 6th, 2021

By the end of first week after the project release you must form the project groups. Five member groups are preferable. Four member groups are acceptable. Three or six member groups are NOT acceptable. You must form groups in LMS through the Project Groups page by self-enrolling in a group. You can use the Discussion Forum to recruit additional team members or ask to join a team – do not join a group without asking, as you cannot later change your enrolment.

Milestone Two - deadline Monday, October 4th, 2021

By the due date you must upload in LMS the final report, the code and a 4 minutes video which explains the content of the report and the obtained results. Please upload a video file directly to LMS, do not upload to YouTube or other video sharing platforms with the idea of submitting a link. You do not need to shoot in 4K or high bitrate modes (the file size should be kept small as uploads to LMS are slow).

Milestone Three - deadline Monday, October 11th, 2021

You have one week after the deadline for report submission to perform peer assessment in Spark^{PLUS}.

Guide for Project Work

The project work is an important component of this course. Working in teams on projects prepares you for professional work in any engineering discipline.

The most important unpredictable factor that affects nearly all engineering work is human behavior, particularly behavior within an engineering team. Learning how to manage this factor is the key to a successful engineering career. This project is intended to help you with necessary practical experience in team skills.

You are free to form groups as you wish. You need to remember that the main constraint is your individual timetable: you must be able to spend time working with your fellow team members.

1. Project Assessment

Your project work will be assessed based on your group's report, code and presentation video. Your team mark will be translated into individual marks for each team member using the peer assessment provided in Spark^{PLUS} as

Individual mark = team mark * Individual's RPF (Relative Performance Factor) to a

maximum of 20.

The team mark for the project will be made up as follows (20 available marks, or 20% of the unit):

Project final report: 5%

Main (group) report as described on Page 7 of this document.

Code 5%

<u>The code must be well documented.</u> A reasonably computer-literate person should be able to run your code without your help.
A person reasonably knowledgeable in theory of mechanisms
should have no problem understanding what your code does.

Presentation Video 10%

The video is a visual presentation of the content of your report. The weighting on the video is high because a good video will save the marker time.

2. Spark^{PLUS}

This unit's group work gives you the opportunity to foster your personal development, realizing critical skills for future employment. One aim of this group work is to produce an effective report. This is achieved through the combined talents of group members, contributing knowledge, skills, and ideas. Another aim is to support your personal growth by an extensive evaluation and feedback scheme. By comparing self and peer evaluations you become aware of your strengths and identify areas for improvement. The evaluation scheme also promotes fairness in marking the group assignments as individual contributions are considered when computing the grade.

How to access it

In this unit you will use a web-based tool called Spark^{PLUS} to confidentially rate your own and your peers' contributions to the assignments.

Link to access Spark^{PLUS}: https://uwa.sparkplus.com.au/login.php

You will be able to provide the feedback in Spark^{PLUS} for one week after the report submission deadline.

Based on a series of answers from each team member Spark^{PLUS} automatically produces two weighting factors: your RPF and SAPA.

RPF Factor

The RPF or Relative Performance Factor is a measure of how the team overall viewed the contribution of each member to the team. This factor will be used to adjust team marks into individual marks using the following formula:

Individual mark = team mark * individual's RPF

Your RPF score can range from 0.8 to 1.2.

SAPA Factor

The second factor calculated is the SAPA factor. This is the ratio of a student's own self- assessment rating compared to the average rating of their contribution by their peers. It provides a student with feedback about how the rest of the team perceives the individual student's contribution.

For example, a SAPA factor greater than 1 means that a student has rated their own performance higher than they were rated by their peers. Conversely, a SAPA factor less than 1 means that a student has rated their own performance lower than they were rated by their peers.

Why are we using Spark^{PLUS}?

Spark^{PLUS} not only makes team work fairer but also encourages the development of professional skills. These skills include giving and receiving positive and negative feedback, resolving conflict, collaborating, assessing your work and the work of your peers, and developing your professional judgment. These are important graduate attributes which are part of the learning outcomes for your degree and are also important to employers. Full participation will enhance both your learning outcomes and your team experience.

How are we using Spark^{PLUS}?

All students are expected to fully participate in this task and required to submit valid assessments. Students who do not complete the task will automatically receive a RPF factor of 0.8.

At all times the unit coordinator retains the right to exercise discretion in relation to application of the RPF factor to the final team mark for all team members. This discretion may be exercised particularly in situations where, in the opinion of the unit coordinator, a team member(s) has / have inadvertently or intentionally, misused Spark^{PLUS}.

Objections

Initially the released RPF and SAPA factors will be preliminary and only become official after any protests are considered. Any student believing their SPARK assessments were unfair may lodge an objection. Any objection to your assessment ratings must be made in writing. Each objection must be max of 500 words clearly outlining why you believe your rating is unfair. Your protest will be reviewed and may be discussed with the other members of your team. Objections must be lodged within 3 days from the date that the Spark^{PLUS} assessments are released.

The lodgment of an objection will be considered as a request for reassessment of the entire team. Hence if a student lodges an objection the marks for the entire team will be reassessed and released after the objection has been considered. In considering any objection the project diary and/or meeting minutes for a team will be reviewed.

3. Technical Project Diary

It is suggested that you meet as a group for at least 1 hour per week and keep track of all meetings using the online Group Diary (blog tool). **This will be your only way to demonstrate your individual contribution to the project if you are unhappy with the feedback received from your peers.** It is not necessary to include this diary in the final report but including the diary in the appendices is encouraged. Upon request your group must be able to demonstrate that you were keeping a diary.

Why keep a diary?

First, your diary will record all your ideas, code specification notes, implementation notes and test results. This is normal professional practice. All professional engineers must work on the assumption that a replacement engineer may need to take over their work at short notice. No one is indispensable in any properly managed engineering project. Your diary is the key to this. It will enable another person to take over your work at any time.

Second, your diary will record the time you spent working at every aspect of this project.

Third, your diary will help you remember your mistakes and successes. Many ideas developed for a project end up being discarded. Sometimes they are not feasible, sometimes they are too expensive. However, they can come in useful in later projects so you need to keep a record.

Ideas

Write your ideas down as they occur to you. Write the date and time at the top of every page or the start of a new section. Your ideas will develop with time: keeping records helps this development.

Implementation Notes

Record names of source code modules and record changes made as you write the code. Changes that do not reflect the specification are unfortunate, but inevitable while learning. Note these changes and variations on the original specification.

Test Results

For software, write test modules that generate the test data inputs and display the results. Record the source files for these test modules in your diary and keep backup copies. Note test results in your diary. If the tests fail and the reason is not immediately apparent, note the results and print out your source code. Leave the problem diagnosis for a later session. Instead of trying to fix the problem immediately, get on with another code module.

Some hours, or days later, return to the problem. Review the test results and follow through the design (or code) to understand why the tests returned unexpected results. If the problem is not apparent, don't keep trying. Design extra displays of the intermediate results and run the tests again. Compare these results with your predictions. By following this routine, you will spend minimal time diagnosing problems.

Is it worth all the time to write this down? Surely it is quicker just to get on with the work!

As an engineer, your skills may be needed elsewhere for something more important at a moment's notice. Therefore, it is essential that you document your work so someone else can take over when needed. Also, you need to prepare for future maintenance and enhancements. Documenting what you do is essential if you want to come back and improve it later. All this is 100% authentic engineering practice.

Importance of a Diary in Commercial Engineering Practice

In a commercial setting, your diary records would form the basis for auditing project accounts and expenditure records. If you have no evidence that you have spent time working on a specific project, your client may not pay you for the work completed. The records of design decisions, and testing results provide a *traceable* record of design and testing that forms an essential part of any quality assurance program.

Final Report

This report will be 10 pages max (12pt font, 1.5 lines spacing) and contain all the information, images, drawings, and results. Your code must be attached to the report (not included in the page limit).

You will need to include:

- Project title
- Group name and members of group (with student IDs)
- Introduction
- Design Process and Methods
- Results and Final Design (dimensions of the mechanism, theta2min, theta2max and the coupler curve generated by your program)
- Discussion/Conclusion

A single report is required from each group of students. You will submit this online through LMS.

Other Important Issues

Back-Up Your Work

It is your responsibility to back-up your own computer work, both in the university laboratories and at home. One back-up should be taken daily: the other should be taken weekly and kept in a safe location. Failure to keep adequate back-ups will not be allowed as an excuse for late work.

Sharing Designs and Software

We encourage students to share design work and software. This is normal industrial practice. However, as the copyright always rests with the original author, the permission of the original author must be obtained and the work of the original author must be clearly marked. The original author must be clearly identified everywhere it is used: in code comments, in the final report, in the list of references. A precise reference is required giving the original source of the code, not just the name of the person from whom it was obtained.

However, all software code must be thoroughly tested, and the testing fully documented in the technical project diary and your report.