Using Classical Mechanism Concepts to Motivate Modern Mechanism Analysis and Synthesis Methods

Robert LeMaster, Ph.D. [1]

Abstract

This paper describes a methodology by which fundamental concepts in the study of mechanisms are used to introduce students to a broad range of topics ranging from the classical graphical methods to state-of-the-art multi-body software. In this method, two central concepts associated with multi-body simulation software are used as unifying threads. The central concepts are: 1) the role of constraint equations, and 2) the difference between inverse-dynamic and dynamic analyses. These central concepts allow a variety of methods to be built on each other in a progressive and unified approach. Students taking a course based on this method are exposed to state-of-the-art technology without sacrificing traditional course content.

Introduction

Most undergraduate curricula in mechanical engineering include a one-semester course in kinematics and dynamics of machinery. The intent of this first course is to teach students how concepts learned in a prerequisite dynamics course can be applied to the analysis and design of mechanisms or machine components. Traditionally, this first course has included an introduction to the kinematic and kinetic analysis of mechanisms using graphical methods.

In recent years, mechanism simulation software has also been used in these courses. The simulation software can be classified into two different types. The first type is limited to a particular type of mechanism (e.g. four-bar, slider-crank, etc), and is usually distributed with textbooks [1]. This type of software can usually animate the motion of the mechanism, and enables students to quickly observe the effects of changing various parameters. These programs are very easy to use and have good user interfaces. The major drawback with programs dedicated to a particular type of mechanism is that they do not expose students to the modeling concepts used by the general-purpose software found in industry.

The second type of software is general-purpose multi-body simulation software. This powerful software enables engineers to simulate complex mechanical systems accurately and efficiently. The graphical user interfaces integrated with these programs enable complex geometries to be developed quickly, and the animation of simulation results provides tremendous insight into the interaction of the different components. The multi-body simulation programs commonly used in the United States includes ADAMS, DADS, and

[1] Department of Engineering, College of Engineering and Natural Sciences, University of Tennessee at Martin, Matin, TN.

Working Model. In addition to these programs, popular CAD programs such as CATIA, I-DEAS, ProEngineer, and Unigraphics, have multi-body modeling and simulation options.

Multi-body simulation software represents the state-of-the-art in the analysis and synthesis of mechanical systems. The theoretical development of the equations used in these programs involves the development and solution of a complex set of differential-algebraic equations. The development of these equations and the study of methods for solving them are typically not covered at the undergraduate level. In fact, courses that focus on the development of the equations, their computer implementation, and solution are taught in graduate level courses, if at all.

Since employers expect students to have familiarity with the tools and methods used in industry, it is important to expose students to state-of-the-art tools and methods. There is a danger in this exposure, in that learning to run a specific simulation program is a skill that can be accomplished with limited understanding of the underlying mathematics or methods used by the program. Therefore, if too much emphasis has been placed on learning how to run a simulation program, the students will have acquired a skill, but will have gained little knowledge. The challenge faced by instructors teaching a first course in the kinematics and dynamics of machines is how to best introduce students to the state-of-the art technology, while at the same time ensuring that they have a grasp on the fundamental concepts, methods, and equations traditionally taught in undergraduate mechanism analysis and design courses.

The purpose of this paper is to present an instructional approach that uses classical mechanism concepts to motivate and provide insight into the workings of state-of-the-art programs. It identifies concepts that are central to both classical and computer based methods, and describes how emphasis is placed on a few central concepts as students progressively move through classical graphical methods, MATLAB simulations, and multi-body simulations using I-DEAS.

There are two central concepts that the author emphasizes to students taking a first course in kinematics/dynamics of machines. These central concepts provide a common theme that tie seemingly different methods or topics together as the course progresses. The motivation for these central concepts is presented in the following paragraphs.

Central Concepts

The algebraic-differential equations that control the motion of a mechanism consist of two parts: 1) the nonlinear algebraic constraint equations, and 2) the differential equations of motion. These algebraic and differential equations are coupled and various solution methods can be employed in their solution [2]. Two distinct types of problems can be encountered during the study of mechanisms: 1) problems requiring the direct solution of the algebraic-differential equations, and 2) problems requiring the solution of the kinematic constraint equations prior to computing the forces necessary to cause the motion. Huag [2] refers to the first type of problem as a dynamic problem, while the second type is called an inverse-dynamic problem. Multi-body simulation programs generally have the ability to solve either type of problem. Therefore, the ability to recognize the type of problem being solved is a "central concept".

Whether a dynamic or inverse-dynamic problem is being solved, constraint equations must be enforced as the mechanism moves. Therefore, constraint equations represent the second "central concept". As we will see, traditional analysis methods that students are introduced to in a first course in kinematics and dynamics of machines use constraint equations quite extensively. However, textbooks rarely mention this unifying or central concept as the various methods are presented, and students see merely a hodge-podge of unrelated topics. In fact, a review of several kinematics textbooks revealed that the word constraint was not used or was used only when discussing the concept of degrees-of-freedom of a mechanism. [1,3-4]

Constraint Equations – A Unifying Thread

The most common feature of all mechanism analysis and design methods is the concept of constraint equations. In their simplest form, constraint equations are algebraic equations that define the position and orientation of all bodies in a mechanism as the mechanism moves. The concept of constraint equations is so fundamental that it can be used as a unifying thread to introduce students to a broad spectrum of topics ranging from graphical methods to multi-body simulation software.

Two types of constraints must be introduced when exposing students to constraint equations. The first type of constraint is a "drive" constraint. This type of constraint gives the position or orientation of a body in the mechanism as a function of time. In the case of a four-bar mechanism, the drive constraint specifies the angular orientation of the drive link as a function of time. The second type of constraint is a "kinematic" constraint. Kinematic constraint equations are mathematical relationships that relate the position and orientation of bodies in a mechanism. The solution of any mechanism problem by any method requires the development of these relationships.

Students are first introduced to constraint equations through the development of "circle-based" constraint equations for a four-bar linkage, Fig. 1. These equations are the simplest form of the equations governing four-bar mechanisms and are the easiest for students to understand. The next step is to introduce students to two important methods for solving these equations. The first solution method is to solve the equations graphically. AutoCAD is used as a tool while performing a graphical solution. As part of the graphical solution method, the Grashof criteria are introduced and are used to demonstrate that there are situations under which solutions of the type desired do not exist.

The graphical position analysis of mechanisms has been included in kinematics courses for many years. The author believes that graphical methods (particularly position based methods) continue to play an important role in the design of mechanisms. The author has observed that in practice, an engineer will often start the design of a mechanism using graphical position analysis methods. These methods often provide the most direct route to identifying a possible design. Once an initial design concept has been achieved, the engineer may move on to more sophisticated analysis and design tools depending on the problem at hand.

The second solution method introduced as a means for solving the "circle-based" equations is the Newton-Raphson method. Students are introduced to the method and are shown how to find the Jacobian matrix. A Matlab program written by the author is examined in detail and used as a demonstration of how to implement the method numerically. The importance of the determinant of the Jacobian matrix in determining whether or not a solution exists is discussed and is related to the Grashof criteria that was developed using graphical methods.

At this point, students have been introduced to the concept of constraint equations and have seen two methods for solving them – graphical and numerical. The concept of solution feasibility has been introduced and demonstrated for both methods. Vector loop equations are next introduced as a method for developing kinematic constraint equations. Instead of locating points at the intersection of circles, vector equations are used to develop the kinematic constraint equations. Position vector loop equations are developed for the four-bar linkage and students are given an assignment that requires them to develop a MATLAB program that solves the vector loop equations using the Newton-Raphson method. This program is very similar to the one developed by the instructor to solve the "circle-based" equations, but involves a different set of equations and associated Jacobian matrix. The results of this exercise are compared to those obtained using the "circle" equations – both numerical and graphical.

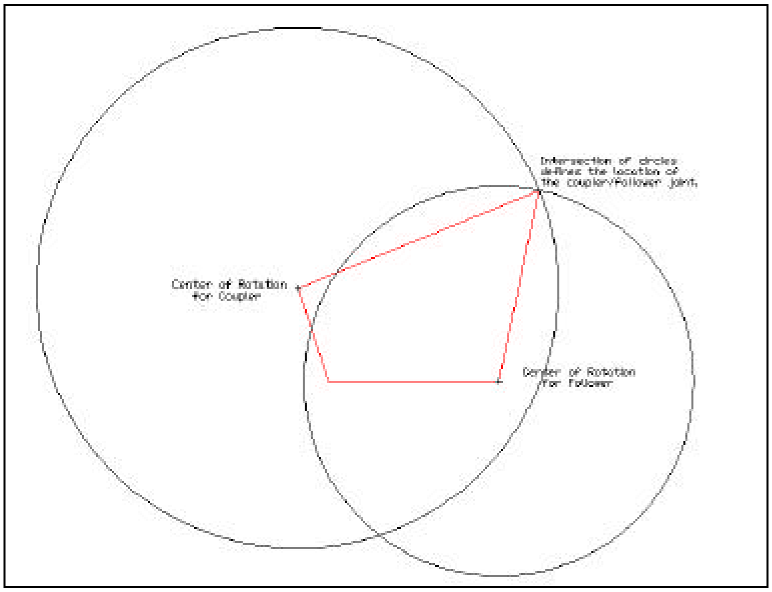

Figure 1. Constraint Circles for Four-bar Mechanism. The coupler and follower joint are always located at the intersection of the circles. The center of rotation of the coupler circle moves with the drive link.

The development of constraint equations for mechanisms other than the four-bar mechanism is next introduced. In-class examples are worked showing how to write and solve constraint equations for mechanisms such as the slider-crank, Geneva wheel, Whitworth Quick-Return, and Swing-Arm Quick-Return. Students are given homework assignments that require them to test their understanding and to develop their skill. In the assignments, students must develop the constraint equations and solve them numerically. They are then required to verify their answers using a graphical position analysis. This dual method approach reinforces the relationship between graphical and numerical solution methods and assists students in trouble-shooting their answers. It also reinforces the importance of anchoring or verifying numerical solutions.

Students initially have difficulty writing the constraint equations, and it is important to follow up these assignments by going over the solutions in class. The author has observed that students struggle with writing the constraint equations because they have trouble visualizing the geometrical relationships involved. The concepts introduced in graphical position analysis are useful in helping students "see" these relationships. As a first step in writing constraint equations, students are instructed to visualize how they would perform a graphical position analysis. Generally, this involves finding the intersection points of lines and/or circles or vector relationships. Next, they are instructed to write down the equations that describe the geometric entities used in performing the graphical position analysis. Once they “see” the geometric constraints that must be satisfied (graphical position analysis), it is easier to write down and solve the equations that describe constraints (numerical position analysis).

Velocity and Acceleration Analysis

The central role of constraint equations is further demonstrated by showing how they are used to find the velocity and acceleration. The concepts of holonomic constraints and generalized coordinates are introduced. These concepts are easily understood at this point because they can be related to what the students have previously seen. In the case of the four-bar mechanism, the generalized coordinates are the angles associated with the vector loop equations or the Cartesian coordinates of the joints in the case of the circle-equations.

The time rate-of-change of the generalized coordinates are found by taking the time derivative of the constraint equations using the chain rule of differentiation. The angular velocities of links and linear velocity of joints are then related to the time rate of change of the generalized coordinates. In a similar manner, the accelerations are found by taking the time rate of change of the velocity equations [2]. Students are required to add the velocity and acceleration equations to the MATLAB program developed during previous assignments, and are required to verify their answers using graphical methods that are also developed during this portion of the course.

Dynamic and Inverse-Dynamic Analysis

The equations of motion for a single link in the mechanism are developed and methods for assembling the equations for a complete mechanism are presented in class. As in similar situations, a MATLAB program developed by the instructor is reviewed and students are required to add the ability to compute joint forces to their programs. These forces are computed using a drive constraint (inverse-dynamic analysis). The difference between a dynamic analysis and inverse-dynamic analysis is experienced by having the students attempt to find the angular velocity of the drive link for a situation in which the torque acting on the drive link is specified instead of the angular position versus time. Once students have determined that they don’t know how or that it can’t be done solving the algebraic constraint equations, the difference between inverse-dynamic analysis and dynamic analysis is explained. Although not examined in detail, Adam-Moulton based methods for solving this type of problem are briefly discussed.

Next, the method for designing a flywheel using a torque-versus-angular position curve is developed. Students use their MATLAB four-bar programs to compute the torque-position curve and design a flywheel. They are then asked to add the flywheel mass and inertia properties to their programs to see the change in torque variation. Since their programs are based on a constant angular velocity constraint equation, they again run into a practical problem that requires a full dynamic analysis (i.e. find the angular velocity of the drive link resulting from an input torque). This process of having students encounter a problem and attempt to explain why what they have learned falls short has proven to be a very good approach for teaching the difference between inverse-dynamic and dynamic analyses. Recall that the ability to differentiate between these two types of analyses was one of the central concepts that the author believes is necessary to understanding the features of multi-body simulation programs.

Multi-body Simulation Software Application

At this point in a course, students have a good understanding of the importance of constraint equations to mechanism analysis and design. They have written constraint equations for a variety of mechanisms and have solved the equations both graphically and numerically. They have seen how the constraint equations are used to find velocities, accelerations, and forces. They also understand the difference between an inverse-dynamic analysis and a dynamic analysis. Students are now introduced to kinematic simulation using the mechanism option in the I-DEAS software marketed by SDRC, Inc.

The instructor demonstrates the use of the software by going through the steps necessary to model and analyze the four-bar mechanism shown in Fig. 2. During this demonstration the equations for a revolute joint are developed and the students see how this type of constraint equation is modeled using the I-DEAS graphical user interface. Students are then given a small design project in which they are required to simulate a practical mechanism using I-DEAS. In their projects they are required to design a mechanism to meet a set of requirements, compute the joint forces associated with the mechanism, and simulate the motion of the mechanism using features within I-DEAS.

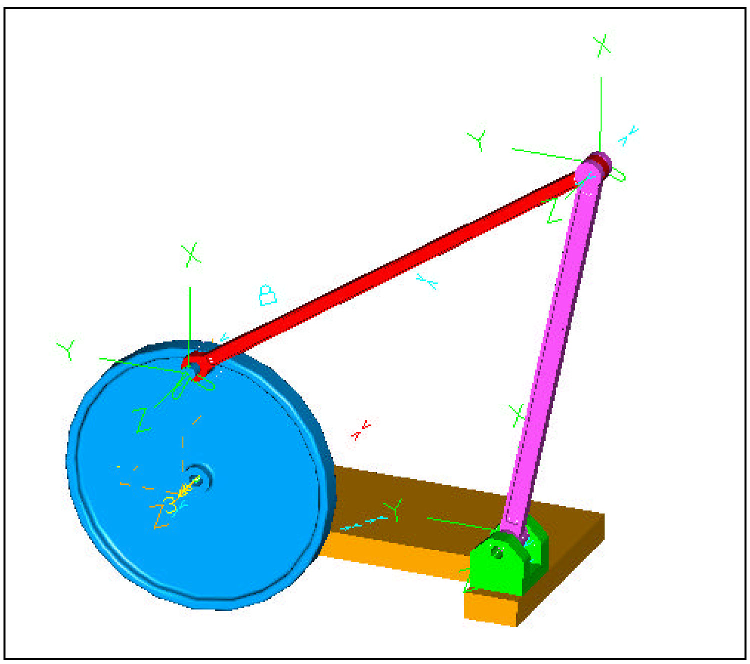

Figure 2. Four-bar Mechanism Modeled Using I-DEAS Software.

Prerequisites

In most kinematics courses the prerequisite is a first course in dynamics. In the case of the course described in this paper, the students should also be able to write computer programs using MATLAB, create drawings of simple mechanisms using AutoCAD, and create models and assemblies using I-DEAS. At the University of Tennessee at Martin, second semester freshmen are introduced to MATLAB and AutoCAD in a second course on Engineering Methods. They are also introduced to part and assembly modeling, and drafting using I-DEAS in a sophomore engineering design course. The differing abilities of students relative to the application of AutoCAD, MATLAB, and I-DEAS is the largest variable in the course, and the instructor must provide considerable assistance as students develop and debug their programs.

Summary

This paper has described an approach for introducing students in a first course on the kinematics and dynamics of machines to the fundamental concepts used in state-of-the-art multi-body simulation software. This approach is based on identifying two central concepts: 1) the difference between inverse-dynamic and dynamic analyses, and 2) constraint equations. The method for using these central concepts as a unifying thread to introduce students to classical concepts as well as modern methods is described. Students taking a course based on this method are exposed to state-of-the-art technology without sacrificing traditional course content.

References

1. Norton, R.L., Design of Machinery, 2nd Ed., McGraw-Hill, 1999.
2. Huag, E.J., Computer Aided Kinematics and Dynamics of Mechanical Systems, Volume 1: Basic Methods, Allyn and Bacon, 1989.
3. Erdman, A.G., G.N. Sandor, Mechanism Design: Analysis and Synthesis, Vol. 1, 3rd Ed., Prentice-Hall, 1997.
4. Waldron, K.J., Kinzel, G.L., Kinematics, Dynamics, and Design of Machinery, John-Wiley & Sons, Inc., 1999.