

Simulation and Animation of Kinematic and Dynamic Machinery Systems with MATLAB

Cole J. Brooking, Donald A. Smith
University of Washington / University of Wyoming

Abstract

This paper describes the application of MATLAB¹ to the problem of determining the motions and forces in kinematic (prescribed motion) and dynamic (prescribed driving force) machinery systems and the animation of the system motions. Depending on the specific information which is required, the machinery system motions are determined from the kinematic loop equations utilizing velocity or acceleration constraints. This treats the motion problem as an ordinary differential equations, initial value problem. For kinematic machinery systems, the reaction force problem is uncoupled from the motion problem. For dynamic machinery systems, the motion and reaction force problems are directly coupled. MATLAB has proven to be a very effective tool for both defining these problems in a mathematical model and for the solution of that model.

In addition to the numerical capabilities of MATLAB, the graphics utilities, and in particular the enhanced handle graphics of MATLAB 5, provide a means of displaying the motions of these machinery systems in the form of a short animation. This animation is very effective in helping the contemporary, visually oriented, student understand the motions associated with machinery systems.

Introduction

Earlier work^{2,3} focused on representing kinematic and dynamic machinery system models as a set of coupled nonlinear ordinary differential equations, and, for specific kinematic systems, the animation of the motion of the system⁴. More recently, several faculty at the U.S. Naval Academy reported the use of MATLAB graphics capabilities to animate aspects of their systems engineering courses⁵. This latter work was the inspiration to extend previous efforts with kinematic and dynamic machinery systems by incorporating the numerical integration of the system differential equations with an animation capability, the major topic of this paper.

The System Model

In order to minimize detail and focus on the animation aspect of the present work a velocity based kinematic simulation of a four bar linkage will be considered. The kinematic skeleton of a typical mechanism is shown in Fig. 1.

This figure is what appears on the monitor for one of the devices simulated and animated with MATLAB. The mechanism is represented mathematically by a set of vectors which form a closed loop. Vector \mathbf{R}_1 goes from the fixed pivot of the output link on the right to the fixed pivot of the input crankshaft on the left. Vector \mathbf{R}_2 goes from the fixed pivot of the crankshaft to the crank-coupler connection. Vector \mathbf{R}_3 represents the coupler and vector \mathbf{R}_4 represents the follower, both of these latter vectors following a generally clockwise loop direction. The position constraint equation is then

$$\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3 + \mathbf{R}_4 = 0 \quad (1)$$

Differentiating the position constraint equation once and taking \mathbf{i} and \mathbf{j} components results in a set of velocity constraint equations which can be used to model the kinematic system and obtain the motion of the linkage configuration. The resultant scalar equations are:

$$\begin{bmatrix} -R_3 \sin(\Theta_3) & -R_4 \sin(\Theta_4) \\ R_3 \cos(\Theta_3) & R_4 \cos(\Theta_4) \end{bmatrix} \begin{Bmatrix} \omega_3 \\ \omega_4 \end{Bmatrix} = \begin{Bmatrix} R_2 \omega_2 \sin(\Theta_2) \\ -R_2 \omega_2 \cos(\Theta_2) \end{Bmatrix} \quad (2)$$

These equations are implemented and solved for the angular velocities ω_3 and ω_4 . The MATLAB function ODE23 uses the low order Runge-Kutta method to integrate the angular velocities to angular displacements. If more accurate results are desired, the higher order Runge-Kutta method could be used by calling ODE45. The resultant angular displacements are used to animate the system.

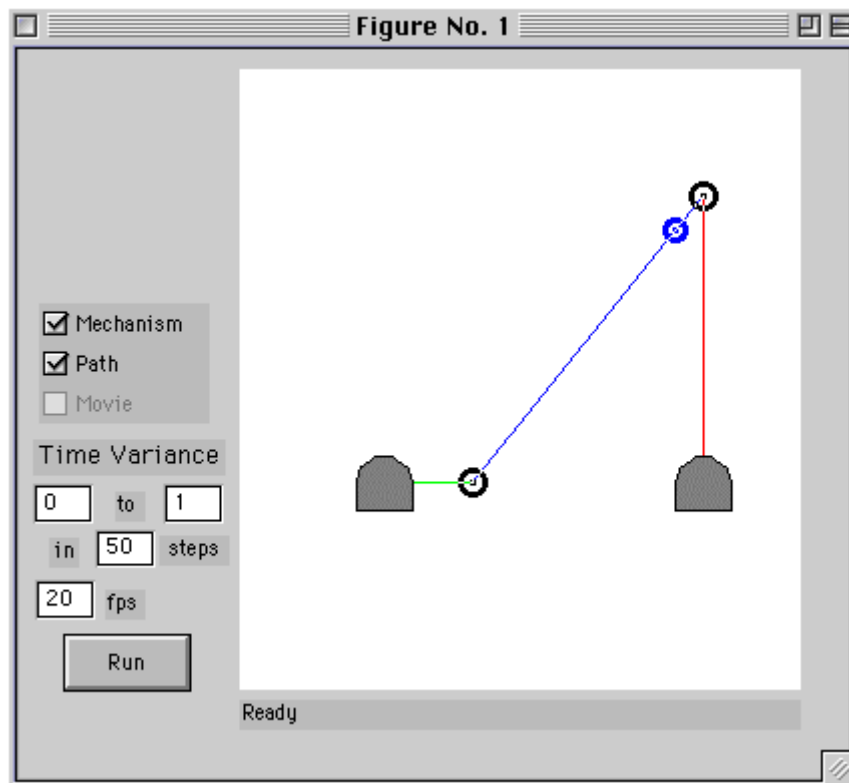


Figure 1: A typically modelled mechanism.

Animation of the Linkage System

Animating the resulting angular displacements is a fairly simple operation using the Handle Graphics routines in MATLAB. A line is defined for each link in the kinematic system. To animate the system, these lines are deleted and redrawn for each time step in the solution, with a delay between each step equal to the inverse of the desired framerate. The lines are created with

an important property: their 'erasemode' is set to 'xor.' This allows MATLAB to redraw only the lines at each time step, significantly reducing the animation flicker and increasing the maximum available framerate.

While this animation may be a fairly straightforward task for an experienced MATLAB programmer to implement for a single kinematic mechanism, it would be quite an ordeal for a student studying kinematics. To minimize this additional effort, the advanced properties of MATLAB Handle Graphics have been utilized to create a GUI for solving and animating various kinematic systems, given the differential equations of motion for each system. The user can specify the initial and final times of the solution, the number of time steps involved, and the framerate at which the solution will be displayed. The dimensions of the links in the model and their positions can also be varied via a simple 'click-and-drag' interface within the simulation window. This allows the student to experiment with different designs and to visualize their effects quickly.

In addition to allowing variations in the mechanism itself, the GUI allows the user to choose a point anywhere along the moving links. The path this point moves through can then be traced as the animation progresses. This allows for concrete visualization of the complex motions involved.

Conclusion and Software Availability

A MATLAB based analysis and animation package for use in a kinematics and dynamics of machinery course has been reported. It is anticipated that the animation capabilities will appeal to contemporary students, typically more attuned to a visual learning mode than their predecessors. The software has considerable flexibility, allowing the student to explore "what if" scenarios by changing link lengths and monitoring the motion of points on the individual links.

The MATLAB code reported here can be downloaded from <http://www23.pair.com/epsilon3/kin/>. It is suggested that the seven downloaded files be placed in a folder inside the MATLAB folder. In order to run the program, extend the MATLAB path to include the folder where the code resides and enter kinimate from the MATLAB control window. On some machines it is advantageous to precede the kinimate command with clear all and set(0,'units','points').

References

- [1] The Math Works, Inc., MATLAB Using MATLAB Version 5, Natick, MA., 1997
- [2] Smith, D.A., R.G. Jacquot & D.L. Whitman, " Digital Simulation of Kinematic Machinery Systems", The International Journal of Engineering Education, V8 #2, 1992, pp. 132-137
- [3] Smith, D.A., R.G. Jacquot & D.L. Whitman, " Digital Simulation of Dynamic Machinery Systems", The International Journal of Engineering Education, V8 #6, 1992, pp. 461-465
- [4] Smith, D.A., R.G. Jacquot & D.L. Whitman, "A Software Package to Enhance the Teaching of Engineering Dynamics." Computer Applications in Engineering Education, V3 # 7 1995, pp. 21-28

[5] Watkins, J., G. Piper, K. Wedward, & E.E. Mitchell, "Computer Animation: A Visualization Tool for Dynamic System Simulations", Proc. 1997 ASEE Annual Conference, Session 1620, June 1997

Cole J. Brooking is currently a Graduate Student in Mechanical Engineering at the University of Washington. After receiving his BSME from the University of Wyoming, he interned at The Mathworks, Inc., studying Mechanical Engineering applications of MATLAB. His current research work involves shape skeleton based design optimization.

Donald A. Smith is currently a Professor of Mechanical Engineering at the University of Wyoming. He received his BSME and M.Sc. degrees from The Ohio State University and his Ph.D. from the University of Michigan. Prior to joining the University of Wyoming he was employed with Xerox and IBM. His academic interests are in the areas of Mechanical Component Design, Kinematics and Dynamics of Machinery, and Computer Aided Design. Current research activity is focusing on web based mechanical component design.