Int. J. Engng Ed. Vol. 12, No. 5, p. 339–346, 1996
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Dynamic Teaching Solutions*

N. SCOTT
B. J. STONE
Department of Mechanical and Materials Engineering, The University of Western Australia, WA 6907, Australia
D. G. DEVENISH
R. D. ENTWISTLE
Department of Mechanical Engineering, Curtin University of Technology, GPO Box U1987 Perth, WA 6001, Australia

This paper is a description of a first-year engineering dynamics course under development at two universities in Western Australia. The first part of the paper describes software intended to complement the traditional course structure, which is novel in that it consists of a unified set of book-like notes and fairly sophisticated computer animations of relevant dynamic systems. The second part of the paper describes physical laboratory exercises intended to broaden the students' appreciation of the applicability of dynamics. The ultimate aim of this work is to develop a rich learning environment that will delight, challenge and inform its audience.

PART 1: SOFTWARE

Philosophy

THE place of 'teaching' software should be to give students access to something they would otherwise have to laboriously create for themselves, in situations where that would detract from the overall learning experience. A simple example is the use of the electronic calculator. We consider that evaluating a large expression by hand would be so time-consuming and boring that students would lose time and inclination to learn the higher meaning of the expression—so we give them a simple tool to do that quickly and get on with other things.

In the case of engineering dynamics software, the authors have found three main areas in which this principle can be applied: improvements to book-like material, stimulation and assessment.

Improvements to book-like material

Making conceptual connections.

A well-written book provides many different kinds of links among the concepts it presents: a contents page, an index, footnotes, references, appendices and so on. The text may even direct the reader to a particular page to learn of some related matter. But making use of these links requires time, skill and confidence. When presenting new subject matter, links of this sort are therefore a minor obstruction to learning.

In a computer environment it is possible to present 'pages' of information—text and diagrams, which can be accessed in sequence, like those of a book—and also to provide 'buttons' on a particular page that can instantly take you to a different page or another part of the book. In the new location there could be more information, or there could be a simulation or a test, as described below. The important thing is that it is possible to return easily to the original page sequence at the starting point of the digression (see Fig. 1).

For example, suppose there are some pages in such a 'book' devoted to concepts in rectilinear motion. The main page sequence might show that

$$v = \frac{ds}{dt} \tag{1}$$

where v is speed, s is distance and t is time, the point being that speed can be reckoned as the *slope* of a graph of distance versus time; later there might be some pages explaining that

$$a = \frac{dv}{dt} \tag{2}$$

where a is acceleration. By this stage the reader may not remember how speed was defined, may be feeling uncomfortable about the use of the derivative operator, or may not appreciate that a is to v as v is to s. So there could be a button near equation (2), like Definition of speed or What is a derivative?, leading to an appropriate digression, without breaking the conceptual flow of the main page chain.

Step-by-step developments.

Another important advantage of a computer 'book' is that constructions and examples can be visually 'built up' under the student's control. The authors have found that students appreciate this kind of display while they are learning how to construct, for example, the velocity diagram for a

---

* Accepted 28 September 1995.

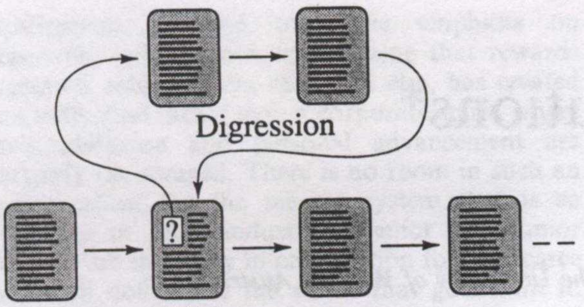


Fig. 1. Links among pages in a computer 'book'.

mechanism. To get them started in drawing their own diagrams, an example problem is shown, and then the student can control a display to which new features are added in sequence, as shown in Fig. 2.

Implementation of book-like software

The authors currently use HyperCard™ running on the Macintosh® platform. It stores the 'cards' (pages) in one continuous chain, but allows arbitrary moves to any card in the chain. In this way digressions can be accommodated by putting those cards at the end, accessible only by buttons placed in the main chain. Step-by-step developments can be implemented as digressions where each card is actually a duplicate of the previous one, with minor additions.

Much the same effect can be achieved using equivalent commercial software available for the IBM PC platform.

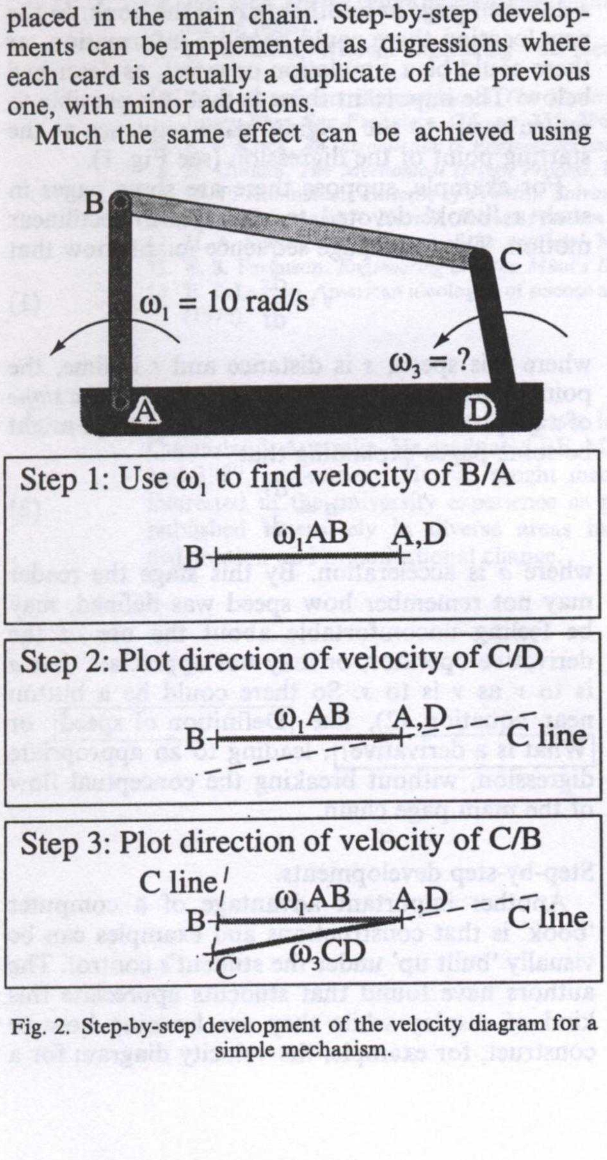


Fig. 2. Step-by-step development of the velocity diagram for a simple mechanism.

Simulation

Numerical software.

There seems to be a trend in educational dynamics software design where the greater part of the student's exposure to the studied system is itself heavily abstracted. A typical example of this sort of software relies heavily on pages of interrelated graphs which can be controlled using buttons and dials on the screen. In this environment the student might encounter the mechanism of Fig. 2 as a static diagram, the only 'animated' displays being plots of ω_3 versus ω_1, angular acceleration of link 3 versus driving link angle, and so on.

There is clearly a place for such software according to the principle given earlier, that anything which frees the student from a boring and non-educational calculation is beneficial; however, it is also important to meet the needs of first-year students for basic visualization.

Visualization using software.

Engineering dynamics is a subject that frequently requires the student to visualize abstract 'objects' such as the components of an acceleration vector. To newcomers the degree of abstraction required can be an obstacle to learning, particularly since in a dynamic system the magnitude and direction of the vectors in question are probably in constant flux.

To address this situation it can be quite helpful to devise illustrative computer animations of particular dynamic systems that show vectors and other abstract constructs superimposed on representations of the solid objects involved. Ideally these simulations should be accessible from the book-like material, perhaps by way of specially marked 'buttons', and thus behave as dynamic illustrations dispersed in the text.

Good simulation software can offer far more than simply a dynamic display: if parameter variation is allowed, students can use it to explore what might actually happen in the real system under a range of conditions. For example, it is all very well to discuss, say, static and kinetic friction; but students have difficulty with problems until they can clearly visualize several key concepts:

1. Under static (sticking) conditions, the frictional force can and will take *any magnitude at all* to match other forces on a body—up to a point;
2. Under kinetic (sliding) conditions there is a prescribed frictional force magnitude $F = \mu_k R$.

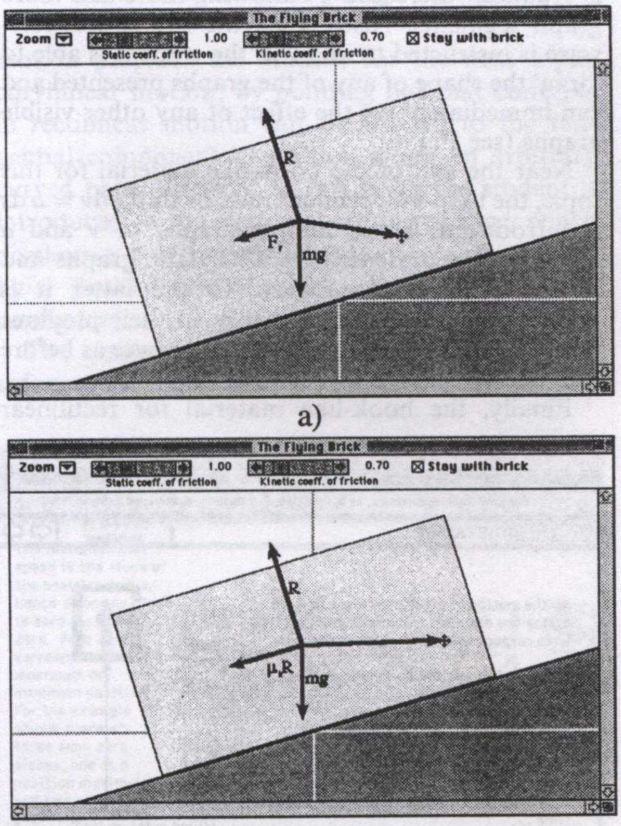
To illustrate the link between these concepts, and appropriate simulation software, consider a package developed by the authors, entitled 'The flying brick', that allows the user to clearly 'see' all the forces on a block that is initially at rest on a slope. The user can

- 'drag' the slope to make it steeper, or
- 'click' on the block with the mouse, visually adding a new force to the block. The magnitude of this 'user' force is, like the others, proportional to its visual length (see Fig. 3).

Several 'experiments' are possible using the flying brick:

1. The user can choose static and kinetic frictional coefficients, and then (a) tip the slope until sliding begins, and (b) find the slope at which sliding proceeds at constant speed;
2. Under static conditions the user can gradually add a force parallel to the ground and clearly see that the static frictional force grows or shrinks to match—until too much is asked of it, at which time the frictional force changes to $\mu_k R$ and the brick begins to slide.
3. If the user applies sufficient force 'upward', or if the slope is actually 'upside-down', the brick does actually 'fly', and it is clear that the only forces acting on it then are gravity and whatever the user is adding. It accelerates through space appropriately—until it comes into contact with the plane again.

The user can view the motion of the brick as though fixed to its centre, or as though fixed to earth. The former is particularly important when performing experiment 2, so that the magnitude of the user's applied force does not change when sliding conditions begin.



a)

b)

Fig. 3. The flying brick (a) with the user adding a force under static conditions and (b) after kinetic conditions have begun.

Assessment

The educational model we work with requires students to sit for examinations. It is well known that the form and content of these examinations largely determine the nature and depth of learning a student will attempt to acquire:

> Most students are 'strategic learners' and so concentrate on the kinds of learning which will give them high marks. 'Assessment tends to draw learning through a course.' [1]

Tutorial interactions are rightly perceived by students to be merely 'rehearsals' for the main examination; therefore many students eagerly await any positive feedback due to them for work handed in, using it to measure their personal progress towards fitness for 'the exam'.

Any delay in this feedback is an impediment to learning, because students quickly move on to new work, and in particular quickly put behind them anything that makes them uncomfortable—such as material they fail to appreciate correctly. This is another area in which a carefully crafted computer environment can help the student: by allowing them to attempt posed problems, and by giving them instant feedback about their assessed state of understanding of the subject matter. Some details of one such package, developed by the authors, are given in Assessing understanding of rectilinear motion.

Implementation of simulation and assessment software

A simple moving illustration that allows no parameter variation is easily implemented in the Macintosh environment by means of a QuickTime^{TM} movie. Such movies can be played directly from HyperCard^{TM} by means of an extension available from Claris Corporation with their commercial development package. The authors have experimented with computer-generated movies of common engineering mechanisms, aimed at students who may never have been exposed to them 'in the flesh'. The result was pleasing, but rather demanding of computer time to generate (see Fig. 4).

Simulation of a dynamic system with parameter variation requires more sophistication. The packages the authors are presenting were developed using Symantec Corporation's THINK Pascal version 4.02 on the Macintosh. Extensive use was made of object-oriented Pascal, allowing many of the simulations to share code for basic functions, leading to a more consistent user interface. The code was compiled into an 'application' or executable file, which could be accessed easily from HyperCard. The ten or so simulations implemented so far share so much code it has proved to be convenient to compile them all into one application, called 'the D100 package'.

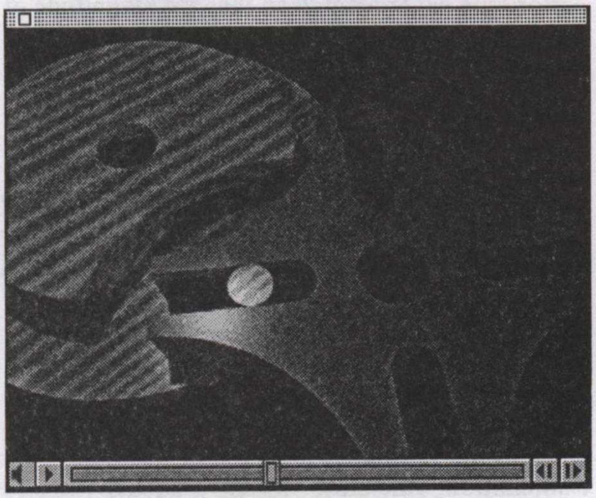


Fig. 4. A still from a computer-generated 'movie' of a Geneva mechanism.

Program linking.

The D100 package is able to receive instructions from HyperCard through a mechanism of inter-program communication that is unique to the Macintosh environment. By this mechanism a custom addition to the standard HyperCard program (an XFCN) can send encoded bytes representing, for example, an increase in a simulation parameter. These requests are received by the D100 package and have the same effect as changes the user might make via controls available to them on the screen (see Fig. 5).

One of the most important ways in which program linking of this sort can help students is to introduce them gradually to the features of each simulation. The first time HyperCard 'calls up' a given simulation it can request that the display be as simple as possible, perhaps just the basic physical layout of the presented objects. Subsequently the student might be shown more: a vector, described in the text, or a new configuration of interest.

Example: rectilinear motion

The software described in this section is intended to complement the lectures and tutorial problems the students encounter when beginning the dynamics course. One of the aims of this first part of the course is to challenge the understanding of dynamics that the students bring with them from high school, which is known to be generally poor. The class also usually contains a proportion of students who have not been exposed to elementary calculus.

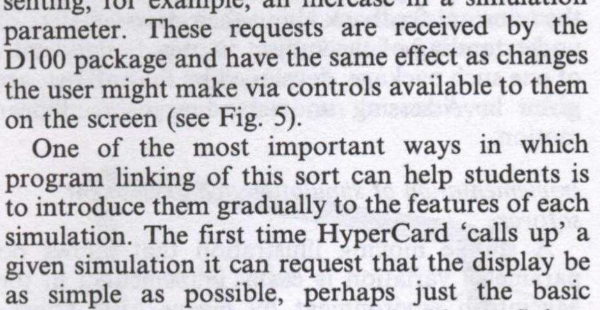


Fig. 5. Parameters can be received by the D100 package from HyperCard$^{\mathrm{TM}}$ or from the student.

Interaction between the 'book' and simulation software.

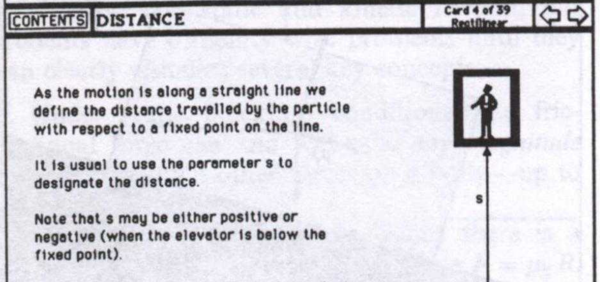
The book-like software for this unit begins very simply by showing the students a preset animation of an elevator moving vertically in its shaft, first in one direction only, and then both up and down. A coordinate is then defined, in this case the s coordinate, which is taken to be zero at the bottom of the shaft and increases with increasing height (see Fig. 6).

The concept of a graph of a motion parameter with time is then presented by showing the student a graph of s against t. Clicking on the button near this graph causes HyperCard to call up a simulation package, OTIZ elevator Co., which displays the *same* graph but animates the motion of the elevator and the position marker on the graph (see Fig. 7).

Subsequent cards introduce the concepts of speed and acceleration, calling the simulation software appropriately at each stage, and then show the student some examples of possible elevator motions. The complexity of these motions is gradually increased by allowing more and more graph segments. Also, when the simulation software is instructed to allow it, the student is able to 'drag' the shape of any of the graphs presented and can immediately see the effect of any other visible graphs (see Fig. 8).

Near the end of the book-like material for this topic, the time-independent relationship $v\,\mathrm{d}v = a\,\mathrm{d}s$ is introduced. Some sample graphs of v and a against s are presented both as static graphs and in the simulation software. In the latter it is possible to also view the graphs in their previous form as functions of time, and to animate as before (see Fig. 9).

Finally, the book-like material for rectilinear



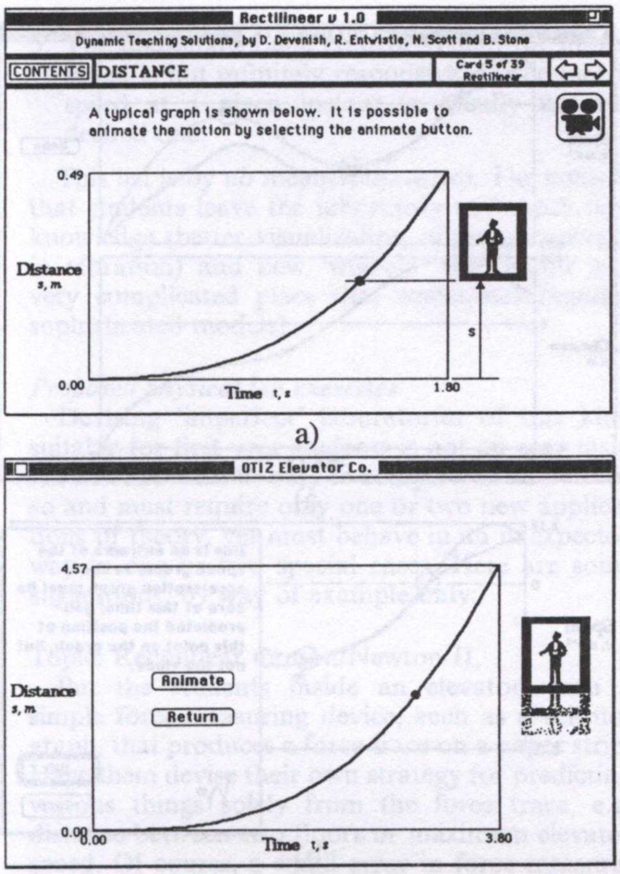
As the motion is along a straight line we define the distance travelled by the particle with respect to a fixed point on the line.

It is usual to use the parameter s to designate the distance.

Note that s may be either positive or negative (when the elevator is below the fixed point).

It is helpful to consider a graph of s against time ......

Fig. 6. An early page in the rectilinear motion 'book'.



A typical graph is shown below. It is possible to animate the motion by selecting the animate button.

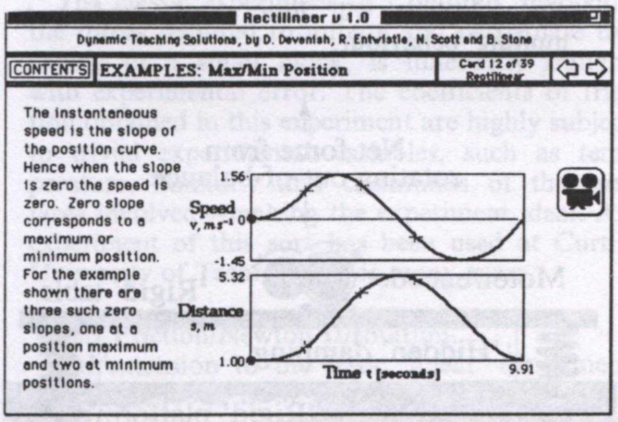
a)

b)

Fig. 7. (a) A static display in HyperCard; (b) the same graph shown in the simulation package, animated.

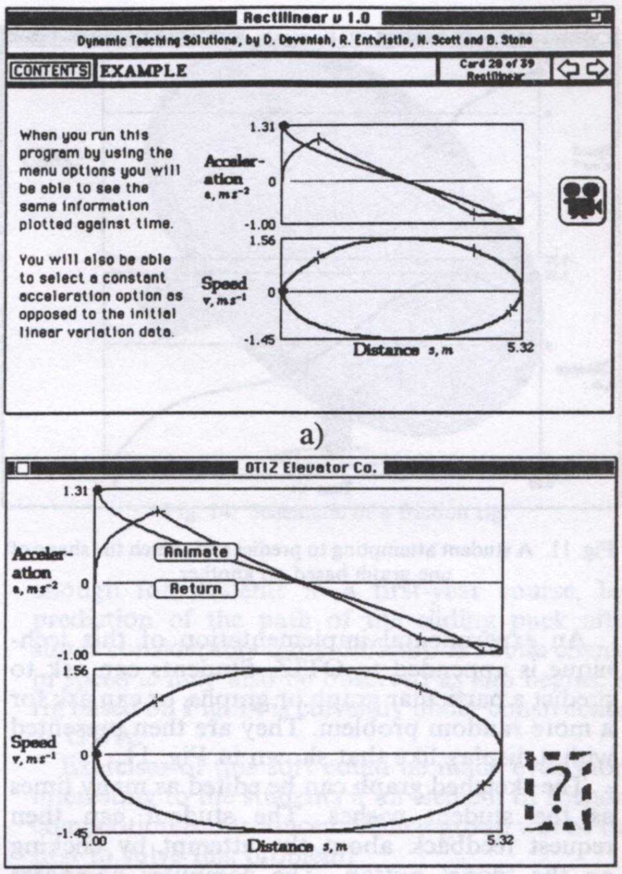
motion prepares the student for the next topic, curvilinear motion, by pointing out that concepts in rectilinear motion can be applied to the tangential component of motion along an arbitrary curved path in space. At this point the student is introduced to the *next* simulation package, which displays a particle moving along a path defined by a Beziér curve: but in a very simple visual configuration (see Fig. 10).

The common educational feature of all this software is the principle mentioned earlier: the student is introduced *gradually* to new concepts by increasing the range of display and interaction options available to them in the simulation packages.



The instantaneous speed is the slope of the position curve. Hence when the slope is zero the speed is zero. Zero slope corresponds to a maximum or minimum position. For the example shown there are three such zero slopes, one at a position maximum and two at minimum positions.

Fig. 8. A card leading to an animated example of a more complex elevator motion.



When you run this program by using the menu options you will be able to see the same information plotted against time.

You will also be able to select a constant acceleration option as opposed to the initial linear variation data.

a)

b)

Fig. 9. (a) A card presenting an example of a time-independent pair of graphs; (b) the simulation software presents the same graphs.

Assessing understanding of rectilinear motion.

In another paper [2] the authors have described a new technique in computer-aided education, where the student attempts to predict the shape of a missing graph.

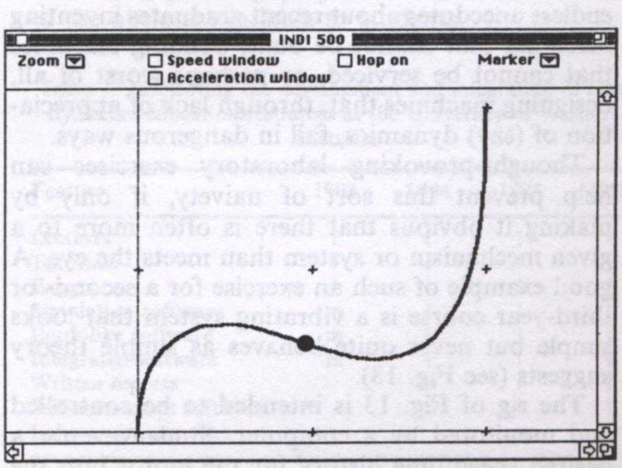


Fig. 10. First contact with the curvilinear motion simulation software.

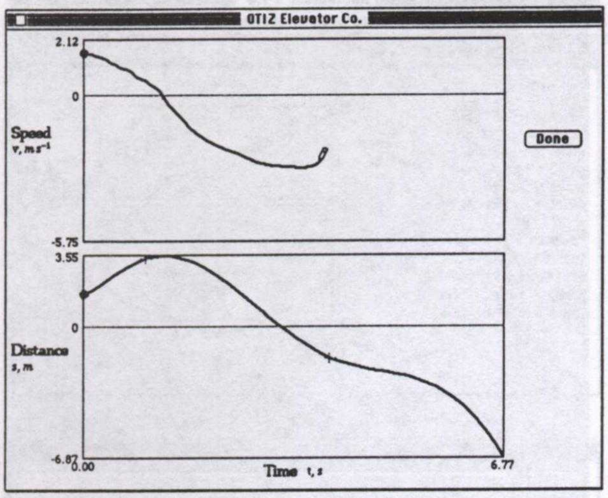


Fig. 11. A student attempting to predict and sketch the shape of one graph based on another.

An experimental implementation of this technique is appended to OTIZ. Students can ask to predict a particular graph or graphs, or can ask for a more random problem. They are then presented with a display like that shown in Fig. 11.

The sketched graph can be edited as many times as the student wishes. The student can then request feedback about the attempt by clicking on the 'done' button. The computer compares their attempt to the ideal solution, using algorithms described elsewhere [3], and gives the student the opportunity to examine a short (but usually positive) critical note about their accuracy at a number of important points (see Fig. 12).

PART 2: PHYSICAL LABORATORY EXERCISES

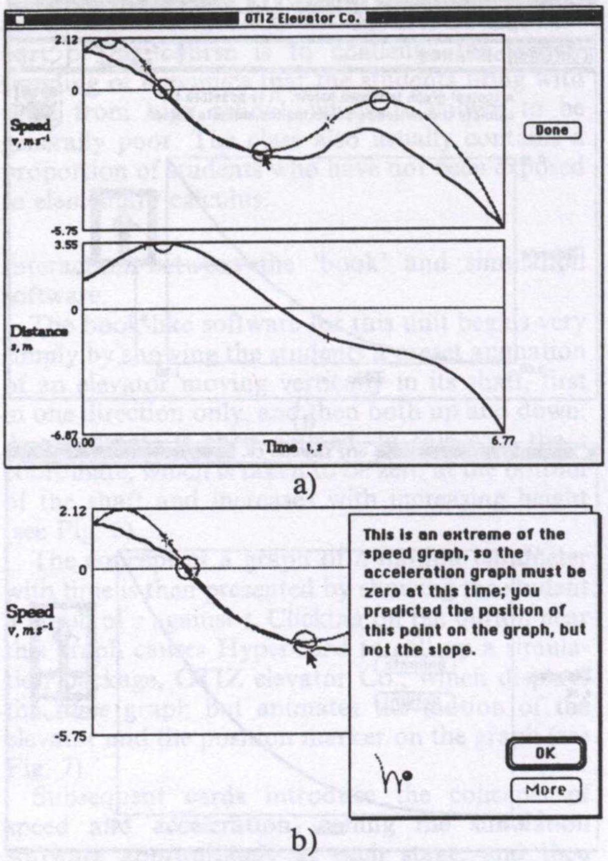
The place of laboratory work

It is a common criticism of professional engineering organizations that graduate engineers lack problem-solving ability and frequently have little knowledge of the 'real world' [4]. There are endless anecdotes about recent graduates inventing machines that cannot be built, building machines that cannot be serviced, or perhaps worst of all, designing machines that, through lack of appreciation of (say) dynamics, fail in dangerous ways.

Thought-provoking laboratory exercises can help prevent this sort of naivety, if only by making it obvious that there is often more to a given mechanism or system than meets the eye. A good example of such an exercise for a second- or third-year course is a vibrating system that looks simple but never quite behaves as simple theory suggests (see Fig. 13).

The rig of Fig. 13 is intended to be controlled and monitored by a computer. Students enter a desired speed/time history for the motor into the computer and can then view the digitally measured amplitude of oscillation of the table as a function of time or motor speed. The graphs so generated will be quite different from those predicted using a simple spring-mass model, because:

1. There is additional compliance in both table and platform.
2. There are a great number of modes of vibration of the system, as the table can rotate about all three axes and is not perfectly rigid.
3. There is 'hidden' damping from the position transducers.
4. The motor is not able to deliver infinite forces to the table or infinite amounts of vibrational energy, so when a major modal frequency is approached it will lose speed, leading to 'hunting' behaviour.



This is an extreme of the speed graph, so the acceleration graph must be zero at this time; you predicted the position of this point on the graph, but not the slope.

Fig. 12. (a) The finished prediction, superimposed on the ideal solution; (b) the feedback dialogue box associated with one of the graph features.

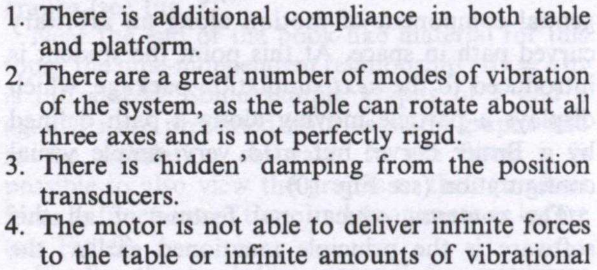


Fig. 13. A deceptively simple rig.

5. The digital/analogue control system for motor speed is not infinitely responsive, so the motor speed at a given instant is usually not the desired one.

This list is by no means exhaustive. The point is that students leave the laboratory with both new knowledge (better visualization of many concepts in vibration) and new 'wisdom' (the world is a very complicated place that sometimes requires sophisticated models).

Proposed physical lab exercises

Devising 'imperfect' laboratories of this kind suitable for first-year students is not an easy task. An exercise must be easy to complete in an hour or so and must require only one or two new applications of theory, yet must behave in an unexpected way in one or two special cases. Here are some suggestions, by way of example only:

Topic: Rectilinear motion/Newton II.

Put the students inside an elevator, with a simple force-measuring device, such as a seismograph, that produces a force trace on a paper strip. Have them devise their own strategy for predicting various things solely from the force trace, e.g. distance between two floors or maximum elevator speed. Of course, a small error in force measurement, or a failure to understand how the force-measuring device works, will lead to poor results. A first-year project to both design suitable force-measuring equipment and perform the elevator measurements has been run at UWA for several years.

Topic: Curvilinear motion.

This follows the same principle as the above, only uses equipment to record the components of force in a moving motor car (which a staff member should drive!). Students could try to predict total distance travelled on the $x-y$ plane, or the radius of curvature of the car's motion at some instant.

Topic: Friction/Newton II.

The classic experiment of Coulomb, involving the forces required to initiate and perpetuate the sliding of a small block, is inherently fraught with experimental error. The coefficients of friction obtained in this experiment are highly subject to trivial experimental variables, such as temperature, humidity and cleanliness of the surfaces involved—making the experiment ideal. An experiment of this sort has been used at Curtin University of Technology for many years.

Topic: Friction/Newton II/rotation.

An extension to the basic 'linear' experiment described above is to examine the motion of a small puck on a rotating (and possibly angled) table (see Fig. 14). The equations that determine the angular velocity for the start of slip are simple enough for students in a first-year course, but prediction of the path of the sliding puck after slip is considerably more difficult. A subtle change in speed ω may also be observed as slip begins. A rig based on Fig. 14 is currently under construction at UWA.

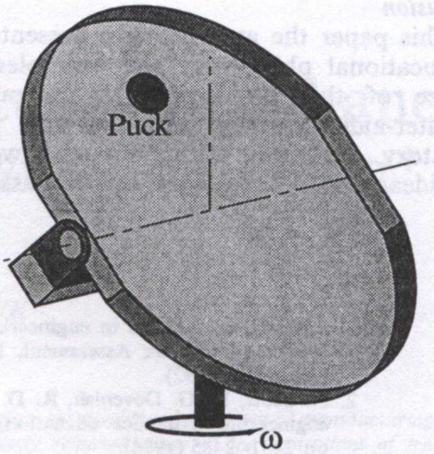


Fig. 14. Schematic of a friction rig.

Exercises of this sort could be made even more interesting to the students if an element of friendly competition is introduced. Which group will be the first to solve this problem?

Course development at UWA

Over the next few years, new course components, such as those described in parts 1 and 2 of this paper, will gradually be added to the traditional course structure at the University of Western Australia, as illustrated in Table 1. Some of the same course changes will also be tried at Curtin University of Technology.

'Integrated software' refers to combinations of book-like and simulation software tailored to the course structure, and intended to be used every few weeks by the entire class.

'Written reports' are required by another first-year subject called Project Engineering 115, but the 'elevator measuring' project described above is one of the options available to students.

Table 1. Illustrating the development and integration of new dynamics education resources at the University of Western Australia

Feature	1993	1994	1995	1996
Lectures	1	1	1	1
Tutorials	1	1	1	1
Exams	1	1	1	1
Simulation software	m	1	1	1
Book-like software	m	1	1	1
Integrated software	m	1	1	1
Written reports		m	1	1
Self-assessment software		m	1	1
Laboratory exercises			m	1

m = used as a demonstration, or in an unfinished form on trial groups of students.

1 = curriculum component.

Conclusion

In this paper the authors have presented both an educational philosophy and examples of the practice of this philosophy in the areas of computer-aided-learning software and physical laboratory exercises. Ultimately the worth of these ideas and practices can only be assessed in retrospect, after several generations of use and development in the actual classroom environment; however, the authors are confident that the approaches given here are likely to make engineering dynamics the most interesting and enjoyable subject for first-year students at the institutions involved.

REFERENCES

1. Assessment methods in engineering degree courses. Discussion document prepared by the EPC Working Party on Assessment, Engineering Professors' Conference Occasional Papers No. 5, December (1992).
2. N. Scott, D. G. Devenish, R. D. Entwistle and B. J. Stone, Computer-based error detection in engineering dynamics education, *Proceedings of ASEE '94 Conference, Edmonton, Canada*, pp. 2481–2485 (1994).
3. N. W. Scott, Simple computer assessment of a sketched graph, *Proceedings ASEE '94 Conference, Edmonton, Canada*, pp. 2571–2574 (1994).
4. The future pattern of 1st degree courses in engineering. Discussion document prepared by the EPC Working Party, Engineering Professors' Conference Occasional Papers No. 3, February (1991).