Int. J. Engng Ed. Vol. 9, No. 4, pp. 322–332, 1993
Printed in Great Britain.

0742–0269/91 $3.00+0.00

A Learning Environment for Robot Mechanics Based on a Hypertext System

E. GIANNOTTI
Dept DIST, University of Genoa, Via Opera Pia 11-A, 16145 Genoa, Italy
C. GALLETTI
Inst. Meccanica Applicata, University of Genoa, Via Opera Pia 15-A, 16145 Genoa, Italy

This paper shows that a programmable hypertext makes it easy to create a computer-based learning environment for mechanical engineering. This allows students to navigate among theoretical topics, industrial examples, and case studies, as well as to perform modelling and simulation activities. The software organization is discussed. As an application, a kinematic analysis of complex robot mechanisms is presented. Users can easily work with the environment, and can also gain a deeper insight into several simulation aspects. Students' reactions to this approach are reported.

1. INTRODUCTION

THE teaching and learning of topics related to innovative sectors of industrial engineering (e.g. robot mechanics and mechatronics) is a complex task in terms of content, methodologies, support tools, and instructional strategies [1]. In traditional education, the teacher's approach consists mainly of lectures for presenting theories, examples and case studies, and of laboratory activities for which students can use professional simulation packages.

Simulation is an important tool for both education and design activities. However, despite the availability of computing facilities, both hardware and software, simulation for education and training purposes in the field of engineering still presents many problems [2, 3]. Several reasons can explain this fact:

- much information on the physical, mathematical, and algorithmic aspects of a problem must be provided to students, due to the complexity of the technology used in innovative sectors;
- the often unfriendly user-interfaces of simulation packages, and the high cost of high-performance and well-interfaced packages;
- the need for software users to understand, at a certain level, the point of view and the mental model of the simulator developer in order to perform a correct simulation;
- the difficulties with using programs that resemble 'black boxes' and cause the user (especially a novice) to make conceptual errors;
- the fact that connections between a simulator and other software facilities (e.g., facilities for storing and retrieving results obtained for different models, and facilities for displaying results in alternative ways, etc.) must be provided when more complex problems are to be solved at a later stage of the educational activity.

In order to teach robot mechanics in a more effective way, we have planned a learning environment to reach three goals: to integrate lectures and laboratory activities in a consistent way; to overcome the drawbacks of industrial simulators; and to offer users a platform in which they can be actively involved. This choice offers the teacher a framework within which he or she can develop and carry out instructional strategies to attain specific learning goals in a systematic way. On the other hand, users may perform learning tasks in a more transparent and autonomous way, arranging their studying and working experiences according to their individual characteristics and skilfulness.

The proposed learning environment exhibits various features:

- possibility of browsing through different knowledge spaces, such as machine theory, industrial applications, case studies, and so on;
- mathematical (kinematic) modelling of a robot by a block-oriented and problem-oriented approach;
- simulation facilities using alternative programming systems and environments;
- numeric and graphic output, with animation;
- various problem-oriented interfaces;
- storing and retrieval of the user's projects;
- interfacing with typical software tools, like graphic generators.

Hypertext technology has been considered appropriate to develop this learning environment. In the past few years, hypertext capabilities in com-

---

* Accepted 4 June 1993.

puter applications have been extensively investigated, and a lot of implementations have been developed [4–7] (e.g. electronic publishing, on-line data bases, software engineering, on-line documentation, computer-assisted instruction, project management, electronic mail, and user interfaces with external devices). By contrast, little attention has been devoted to the possibilities offered by this new technology in the field of engineering simulation and design [7–10]. We show that it is possible to use a hypertext system to integrate different educational activities in a consistent way. Hypertext is self-exploratory and requires a low cognitive overhead, as the user can learn by doing.

To develop the learning environment, we selected HyperCard. Using HyperCard terminology, in this work the term 'stack' corresponds to a hypertext document, and the term 'card' refers to a page of a document that is displayed as a screen view.

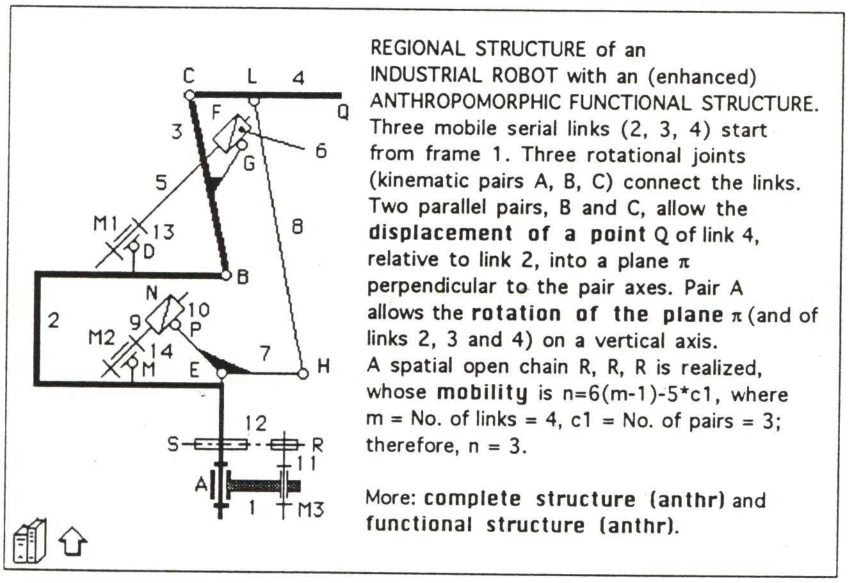
In Section 2 an example of using the learning environment is illustrated to show the main features of this approach as the student can see them. Section 3 outlines the software organization. In Section 4, students' reactions are reported and some remarks on the use of the software are made.

2. USE OF THE LEARNING ENVIRONMENT

This section presents an example of the students' interaction with the material provided by the proposed hypertext environment. All user-program interactions are mouse-driven.

Starting from a menu, the student displays a page containing the sketch of a robot (anthropomorphic) structure and a short text (Fig. 1). The basic functional structure of the robot is enhanced. The text contains standard links (in bold characters) to other pages that provide information related to the current page. The student can decide to know about this typical robot (by clicking the words 'complete structure' or 'functional structure'), or to address the more general problem of obtaining the robot motion ('displacement of a point' or 'rotation of the plane'). For instance, by clicking the words 'displacement of a point', the student can activate a link that displays the kinematic constraints used to move a point in a plane and to give a spatial motion to this plane (Fig. 2). From this page, the student can go back to the original page, or go on via further links. A click on the words 'revolute' or 'prismatic' allows the student to have access to the cards that contain information about mechanism theory. For example, through the link 'revolute', the card in Fig. 3 is shown, with basic theoretical information about the revolute joint. Simple descriptions of the physical and mathematical properties of this joint are presented, and, by clicking the icon of the magnifying lens, a list of Pascal procedures to perform kinematic analysis can be seen. The same information is further referenced by other hypertext documents (see below). Simple animation of the pair motion is allowed.

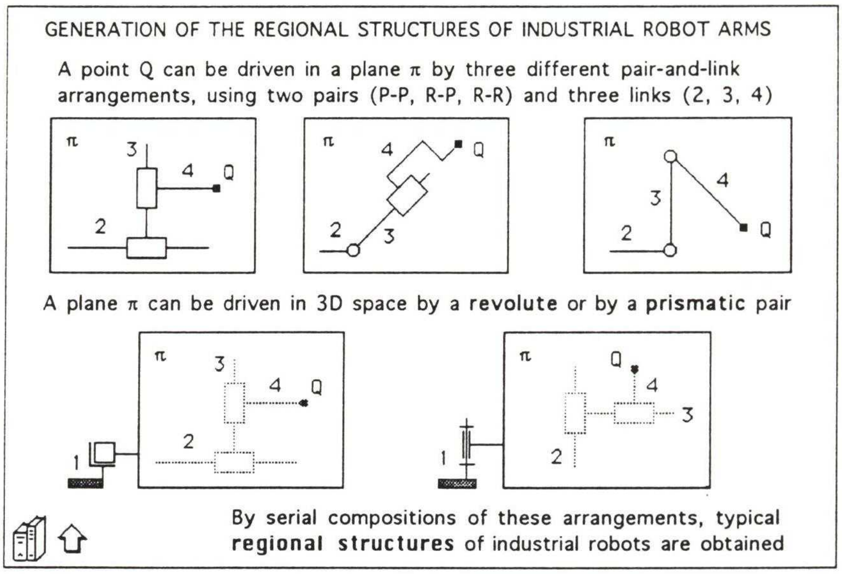
As an alternative, a click on the words 'regional structures' of the card in Fig. 2, makes a page appear that contains commented sketches of all typical regional structures of industrial robots (Fig. 4). Exploded views of all bodies and joints of each robot can then be obtained by clicking the corresponding words. Figure 5 presents a page for an anthropomorphic structure. For each structure element, local reference frames are shown (and can be hidden for better readability, if desired), using a classical notation (HD references) adopted in robot kinematics to describe the body geometry and joint (driver) variables. Note that such views are consistent with the joint definitions given in Fig. 3. Typical values of the HD parameters (s, ϕ,



REGIONAL STRUCTURE of an INDUSTRIAL ROBOT with an (enhanced) ANTHROPOMORPHIC FUNCTIONAL STRUCTURE. Three mobile serial links (2, 3, 4) start from frame 1. Three rotational joints (kinematic pairs A, B, C) connect the links. Two parallel pairs, B and C, allow the **displacement of a point** Q of link 4, relative to link 2, into a plane π perpendicular to the pair axes. Pair A allows the **rotation of the plane** π (and of links 2, 3 and 4) on a vertical axis. A spatial open chain R, R, R is realized, whose **mobility** is n=6(m-1)-5*c1, where m = No. of links = 4, c1 = No. of pairs = 3; therefore, n = 3.

More: **complete structure (anthr)** and **functional structure (anthr)**.

Fig. 1. Structure of an industrial robot.



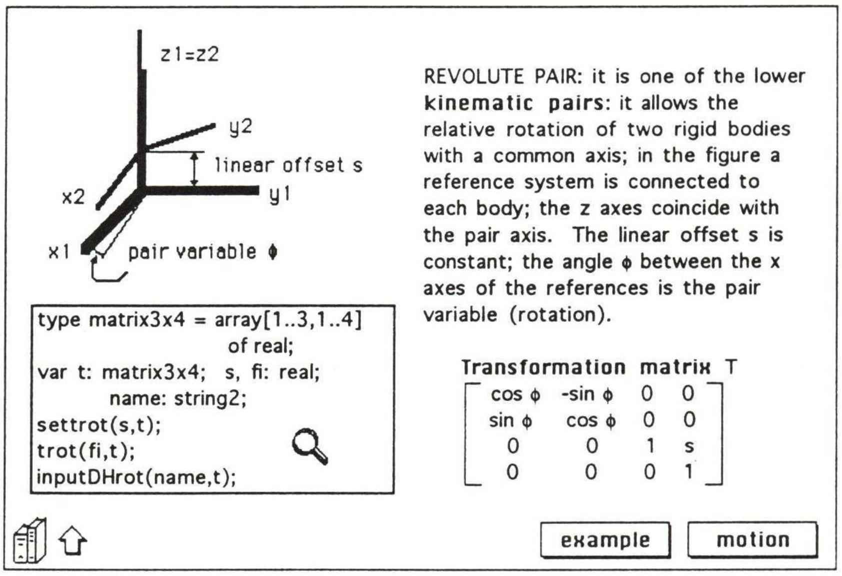
GENERATION OF THE REGIONAL STRUCTURES OF INDUSTRIAL ROBOT ARMS

A point Q can be driven in a plane π by three different pair-and-link arrangements, using two pairs (P-P, R-P, R-R) and three links (2, 3, 4)

A plane π can be driven in 3D space by a **revolute** or by a **prismatic** pair

By serial compositions of these arrangements, typical **regional structures** of industrial robots are obtained

Fig. 2. Constraints for robot motions.



```
type matrix3x4 = array[1..3,1..4]
                    of real;
var t: matrix3x4;  s, fi: real;
        name: string2;
settrot(s,t);
trot(fi,t);
inputDHrot(name,t);
```

REVOLUTE PAIR: it is one of the lower **kinematic pairs**: it allows the relative rotation of two rigid bodies with a common axis; in the figure a reference system is connected to each body; the z axes coincide with the pair axis. The linear offset s is constant; the angle ϕ between the x axes of the references is the pair variable (rotation).

Transformation matrix T

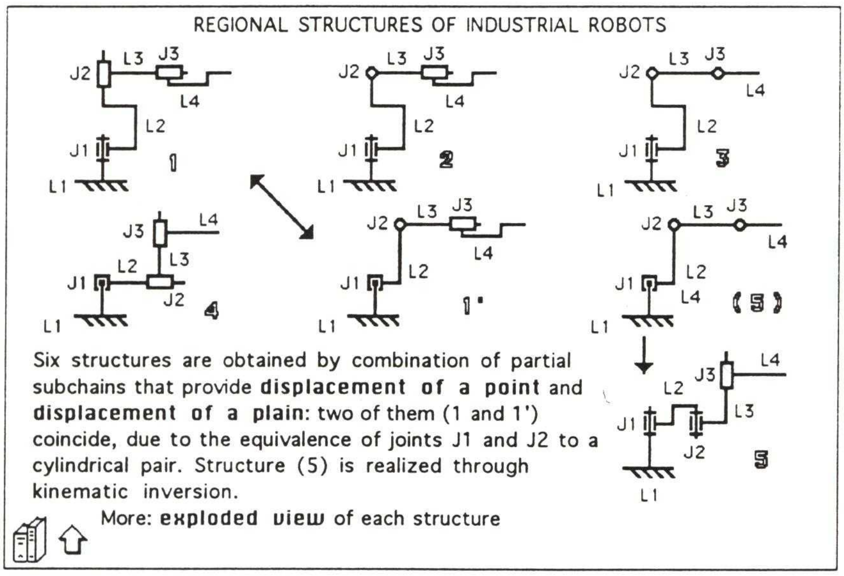
$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & s \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 3. Revolute pair.

a, γ), corresponding to the positions of the bodies displayed in the figure, are also given in this page.

After analyzing this hypertext page, the student can begin a more complex and involving interaction with the material. He can use the system facilities that help him create a mathematical model of the robot, with dimensional data chosen by him. This is done by clicking the button 'entry data'. Now a condensed representation is furnished for each component (joint or link) of the robot (Fig. 6); it includes an abstract sketch of any element and fields for storing the dimensional robot parameters. Some of these parameters depend on the anthropomorphic structure and cannot be varied by the user (locked fields); other parameters are the values of the robot dimensions that can be chosen by the student. Three radio buttons are also provided for choosing a particular arrangement of the robot's terminal body by selecting one of the options i, ii, iii in Fig. 5.

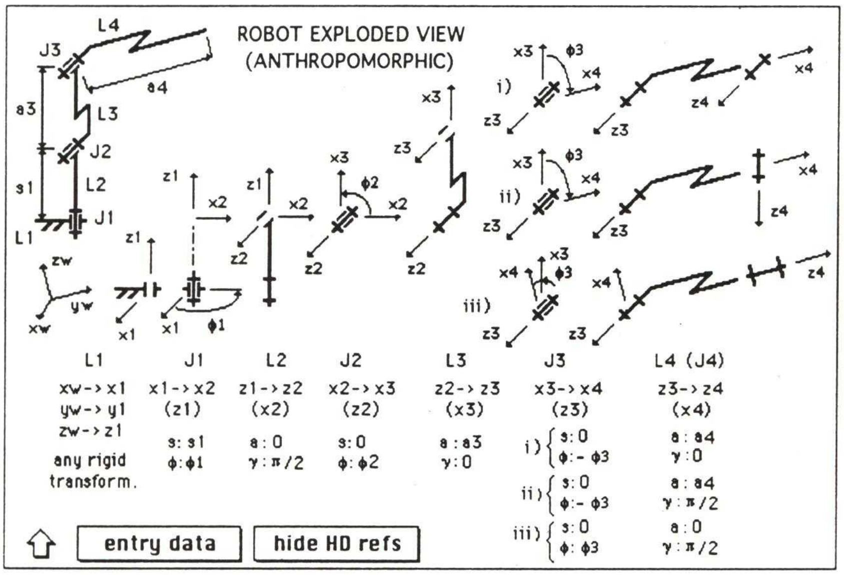
The button 'create model' allows the student to create a new hypertext document containing a mathematical model of the robot. This document consists of a set of cards, one for each component of the robot. For instance, Fig. 7 shows a page corresponding to the third joint (J3) of the robot. Here the hypertext capabilities are fully exploited: in addition to a sketch and to two input-data fields corresponding to those described above, this page contains buttons for activating kinematic analysis algorithms (lower-left corner) and fields for displaying analysis results. Such buttons activate algorithms that compute the coordinate transformation matrices for this joint, using local data



Six structures are obtained by combination of partial subchains that provide **displacement of a point** and **displacement of a plain**: two of them (1 and 1') coincide, due to the equivalence of joints J1 and J2 to a cylindrical pair. Structure (5) is realized through kinematic inversion.

More: **exploded view** of each structure

Fig. 4. Regional structures of industrial robots.



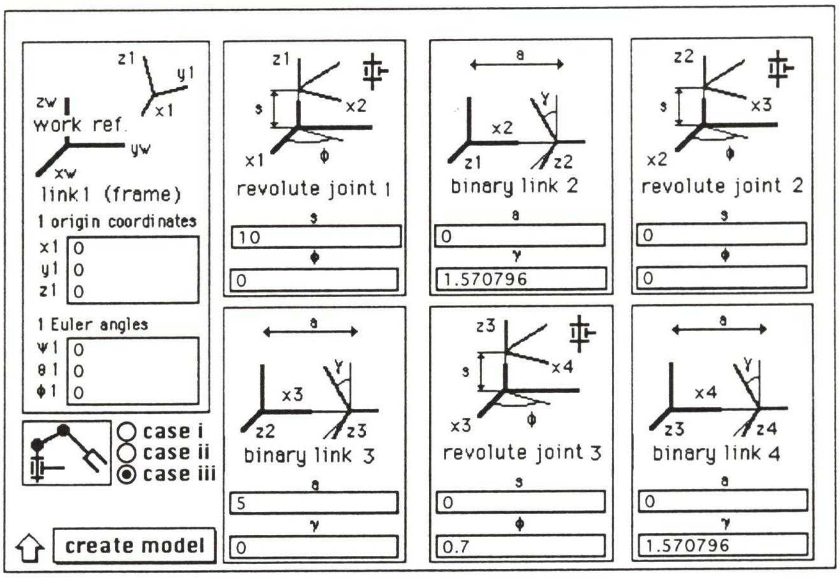
L1	J1	L2	J2	L3	J3	L4 (J4)
xw-> x1 yw-> y1 zw-> z1	x1->x2 (z1)	z1->z2 (x2)	x2->x3 (z2)	z2->z3 (x3)	x3->x4 (z3)	z3->z4 (x4)
any rigid transform.	$s: s1$ $\phi: \phi 1$	$a: 0$ $\gamma: \pi/2$	$s: 0$ $\phi: \phi 2$	$a: a3$ $\gamma: 0$	i) $s: 0$ $\phi: -\phi 3$	$a: a4$ $\gamma: 0$
					ii) $s: 0$ $\phi: -\phi 3$	$a: a4$ $\gamma: \pi/2$
					iii) $s: 0$ $\phi: \phi 3$	$a: 0$ $\gamma: \pi/2$

Fig. 5. Exploded view of a robot structure.

and all the necessary data derived from the component to which the present one is connected. Results of this computation (3 × 4 matrices) are shown in the card fields. These matrices are computed according to the definitions and algorithms given in Fig. 3.

Once the hypertext cards corresponding to the components of the robot model have been generated, a special card is also prepared (Fig. 8) that contains a simple outline of the robot structure (left) and a list of all pages (right). This card is a document index; a click on any element of the sketch, or on any row of the list, makes the pages with the corresponding element appear (for instance, clicking the round rectangle 'J3' makes the card in Fig. 7 appear). Other special cards for controlling the simulation run and the result storage are automatically added to this text. For instance, Fig. 9 presents a card that allows the definition of the motion the robot joints must provide during simulation. By using the fields and buttons on these cards, the student can assign starting, stepping and stopping values to the variables for any robot joint; moreover, he or she can set points and vectors connected to the robot, whose positions he or she wants to be computed and stored during the simulation. Then the model can run in a completely automatic way.

The page in Fig. 8 allows the student to perform more complex operations on the robot model. By using menu-driven options (top line), the student can define interactively any serial robot; then a new document, corresponding to the robot model, is automatically created. Specific cards can be



link1 (frame)

1 origin coordinates
x1 0
y1 0
z1 0

1 Euler angles
ψ1 0
θ1 0
ϕ1 0

○ case i
○ case ii
● case iii

revolute joint 1
s 10
ϕ 0

binary link 2
a 0
γ 1.570796

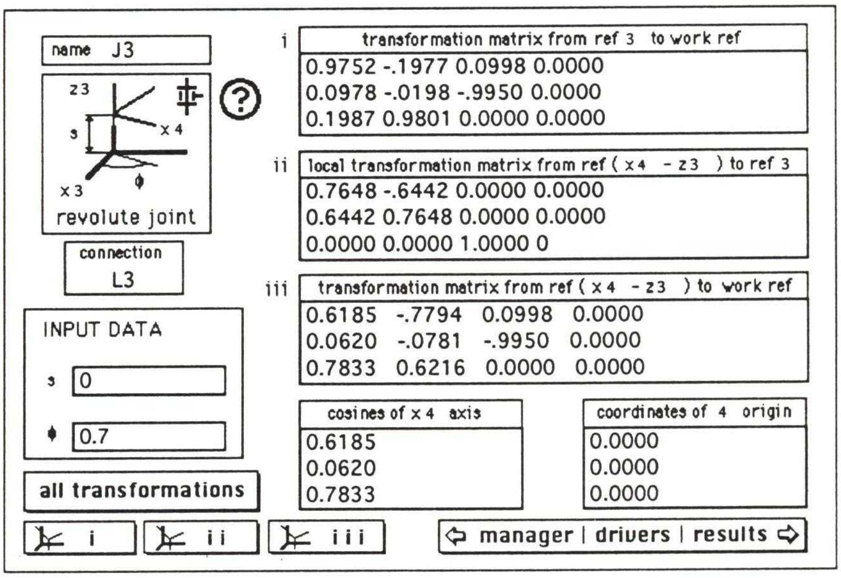
revolute joint 2
s 0
ϕ 0

binary link 3
a 5
γ 0

revolute joint 3
s 0
ϕ 0.7

binary link 4
a 0
γ 1.570796

Fig. 6. Scheme of the model of an anthropomorphic robot.



name J3

connection L3

INPUT DATA
s 0
ϕ 0.7

```
i   transformation matrix from ref 3  to work ref
    0.9752 -.1977 0.0998 0.0000
    0.0978 -.0198 -.9950 0.0000
    0.1987 0.9801 0.0000 0.0000

ii  local transformation matrix from ref ( x4  - z3  ) to ref 3
    0.7648 -.6442 0.0000 0.0000
    0.6442 0.7648 0.0000 0.0000
    0.0000 0.0000 1.0000 0

iii transformation matrix from ref ( x4  - z3  ) to work ref
    0.6185  -.7794  0.0998  0.0000
    0.0620  -.0781  -.9950  0.0000
    0.7833  0.6216  0.0000  0.0000

cosines of x4  axis      coordinates of 4  origin
0.6185                   0.0000
0.0620                   0.0000
0.7833                   0.0000
```

Fig. 7. Model of a revolute pair.

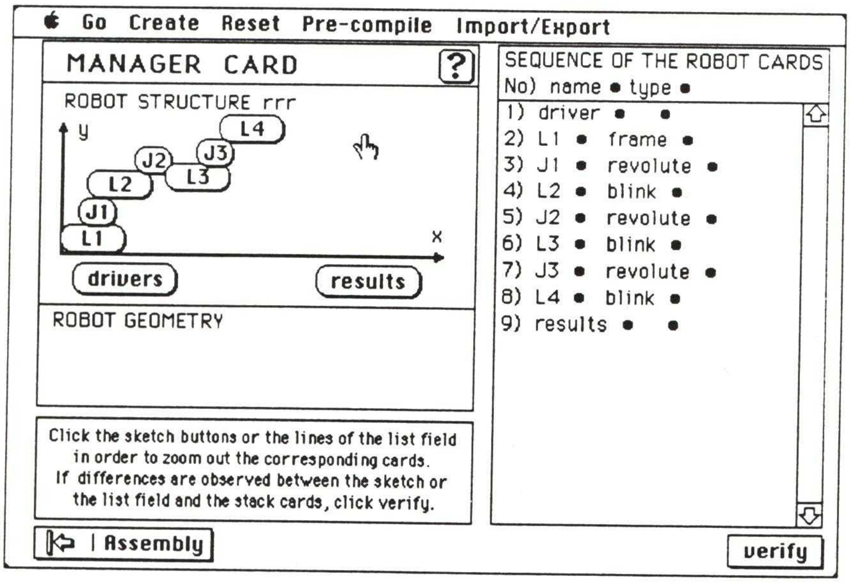
accessed, data are inserted in the related fields, and the simulation is performed, as in the previous case.

As shown in Fig. 1, the complete structures of industrial robots are not always serial and can be made of multiloop chains of links; an analysis of such mechanisms requires more powerful simulation facilities. In order to provide them, multiloop mechanisms must also be defined and simulated by the hypertext system. The way a student can access and use these features is similar to that described above. For instance, Fig. 10 shows a card for defining a planar multiloop mechanism and for managing its simulation. Graphic sketches of the positions of the mechanism can be obtained (Fig. 11).

Finally, the student can ask for the robot structure to be converted into an explicit Pascal code for kinematic analysis (using the 'Pre-compile' menu option in Fig. 8). This code can be examined by the student and exported into a Pascal-computer environment; then, it is translated and linked to Pascal libraries (students can have a look at the Pascal procedures for each mechanical component, as previously mentioned in describing Fig. 3). Then, the simulation run starts, using both the actual data provided by the hypertext cards and the facilities of the Pascal environment.

3. SOFTWARE ORGANIZATION

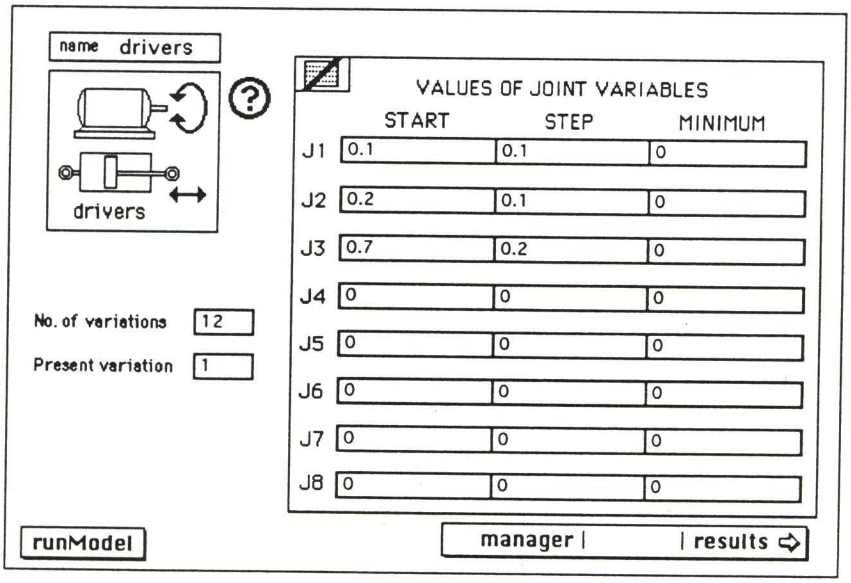
The previous section has shown that two different modalities of using the hypertext system can be adopted. The first corresponds to the standard application of a hypertext, using navigation links



SEQUENCE OF THE ROBOT CARDS
No) name • type •
1) driver • •
2) L1 • frame •
3) J1 • revolute •
4) L2 • blink •
5) J2 • revolute •
6) L3 • blink •
7) J3 • revolute •
8) L4 • blink •
9) results • •

Click the sketch buttons or the lines of the list field in order to zoom out the corresponding cards. If differences are observed between the sketch or the list field and the stack cards, click verify.

Fig. 8. Manager of the robot model generator.



No. of variations 12
Present variation 1

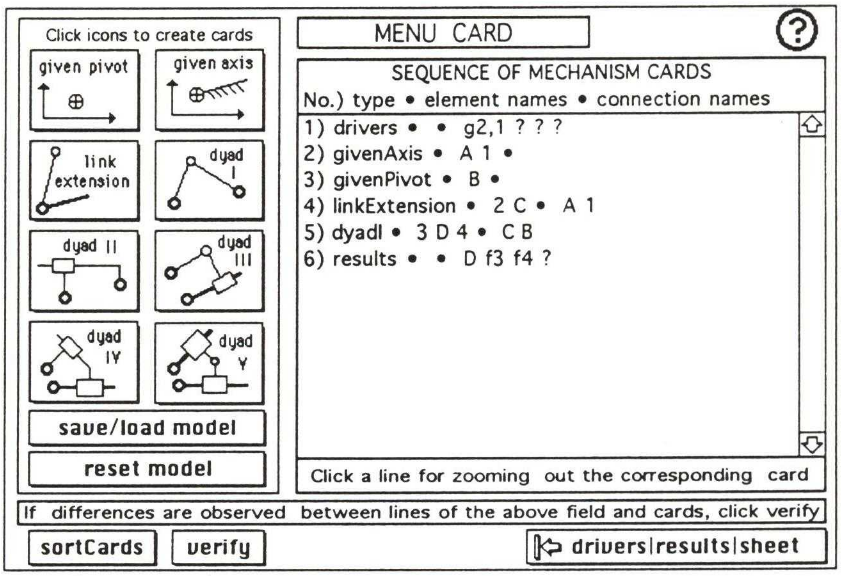
VALUES OF JOINT VARIABLES

	START	STEP	MINIMUM
J1	0.1	0.1	0
J2	0.2	0.1	0
J3	0.7	0.2	0
J4	0	0	0
J5	0	0	0
J6	0	0	0
J7	0	0	0
J8	0	0	0

Fig. 9. Drivers of a robot.

between its cards. Each card makes reference to a short topic related to mechanism theory (e.g. Fig. 3), or to a specific practical example (e.g. Fig. 1).

The second modality is based on a more innovative idea: all the model entities (i.e. elements of the physical model, data structures, and algorithms) are associated with abstract objects [8]. A set of standard cards is prepared for each type of physical robot entity (e.g. joint, link, point, etc.; see an example in Fig. 7). Each card contains fields corresponding to I/O data structures required to simulate that entity, and buttons corresponding to simulation algorithms. Other buttons allow dynamic navigation among the cards when a specific model has been generated. The model of an actual robot is created by choosing from the standard set (through copy-and-paste operations) the correct sequence of pages that corresponds to the robot structure. The page sequence can be created by performing menu-driven interactive operations (via the menu bar at the top of Fig. 8, or the icon button in Fig. 10), or in a predefined automatic way (for instance, starting from the page in Fig. 6). In both cases, for the students' convenience, each card contains names corresponding to the actual physical object it represents. For instance, the page for the third joint of any robot (Fig. 7) presents the correct names of the object itself and of the body the object is connected to. Moreover, the names of transformation matrices and of other output data are written above their data fields, and are automatically made to correspond to correct HD reference systems (see also Fig. 5).



MENU CARD

SEQUENCE OF MECHANISM CARDS

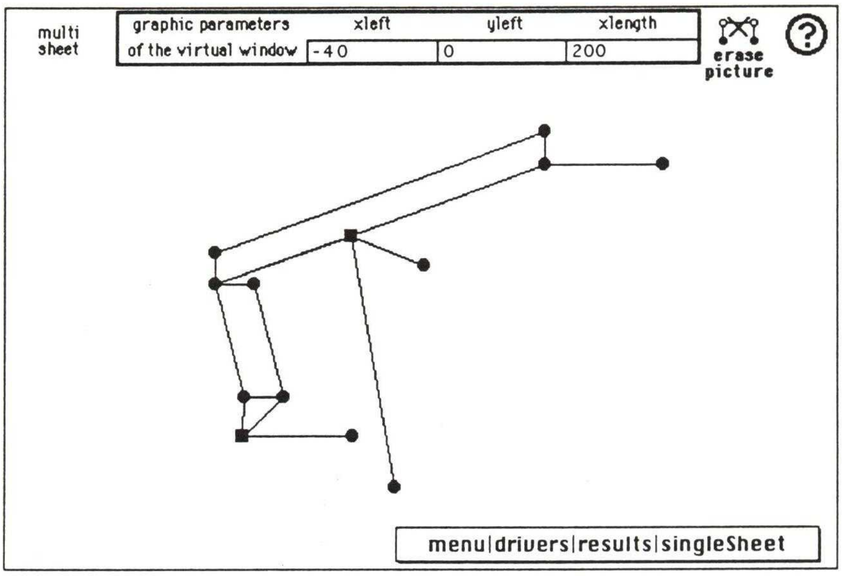
No.) type • element names • connection names

1) drivers • • g2,1 ? ? ?
2) givenAxis • A 1 •
3) givenPivot • B •
4) linkExtension • 2 C • A 1
5) dyadI • 3 D 4 • C B
6) results • • D f3 f4 ?

Click a line for zooming out the corresponding card

If differences are observed between lines of the above field and cards, click verify

Fig. 10. Manager of the planar model generator.



graphic parameters of the virtual window

xleft	yleft	xlength
-40	0	200

Fig. 11. Sketch of a planar mechanism.

Kinematic simulation is performed in the following way: a special card (e.g. Fig. 8) sends to the cards that correspond to the robot model (e.g. Fig. 7) a sequence of messages to trigger the buttons that activate the simulation algorithms (e.g. the button ‘all transformations’ in Fig. 7).

A simplified scheme of the software organization to deal with both modalities is shown in Fig. 12. The core of this organization consists of three hypertext documents: master objects, model generators, and instances.

The master objects visible on the left side of Fig. 12 correspond to the components that are used to define a mechanism; they constitute a document that is prepared, once for all, by the system developers, and, usually it is not modified. The enhanced box in the figure represents the model generators. A model generator is a special hypertext that is accessed by the user to create the model of an actual mechanism and to manage the simulation run. Its basic function is to create the user-defined hypertext that contains the actual mechanism model as a sequence of instances of master objects. Up to now, 2 generators, 1 for planar closed-chain mechanisms and the other for serial spatial robot arms, have been developed. For example, Fig. 8 is the most important card of a model generator for robots and provides the basic interface between the user and the generator; Fig. 7 is an instance of a master object: obviously, in the master object all data fields are empty and reference systems are still undefined.

The choice for the input data in Fig. 7 to be used for a simulation (e.g. ϕ and s) is consistent with

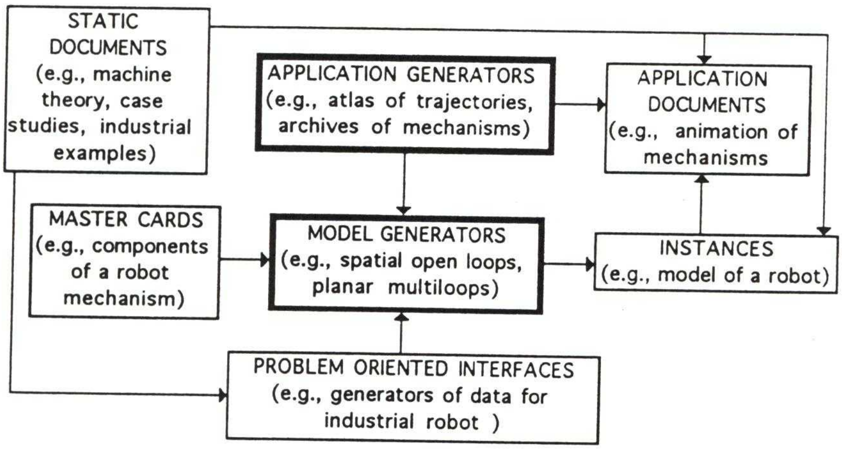


Fig. 12. Software organization.

classic methods of kinematic analysis of spatial mechanisms; such data are suitable for an operator trained in robot kinematics but may be difficult for an untrained user. However, a hypertext system makes it easy to provide a higher-level interface to help the user with data entry. To this end, a document acting as a preprocessor can be realized to allow the user to define the data for typical robot structures in an appropriate way. This is presented in Fig. 12 by the box 'problem oriented interfaces'; examples are given in Figs 5 and 6.

The 'application generators' (Fig. 12) are other hypertext documents used to create special application documents. They invoke the model generators to manage a specific mechanism model, and build documents with application-oriented cards. For instance, a generator of trajectory atlases for mechanisms has been implemented. The user can access the atlas generator and define the range of the mechanism dimensions to be varied. Then, an atlas is automatically created by sending appropriate messages to the model generator and by storing its graphic results. When a specific atlas document has been created, the hypertext capabilities are used to perform an interactive nonlinear navigation through it. For instance, Fig. 13 shows a page of an atlas of trajectories for 4-bar linkages that has been generated in this way. Each card contains navigation buttons that allow the user to display other atlas pages in increasing or decreasing order of the variations in each dimensional parameter (a, b, etc.).

In a similar way, new documents containing cards with sketches of mechanisms in different positions can be generated and retrieved, thus allowing an animation effect.

The consistency of all these different documents in the proposed hypertext environment makes it easy to connect them and to make them work together. In any case, all generated documents (models, atlases, mechanism libraries, etc.) can be examined, copied, pasted, modified, and improved by users.

It should be noted that many interesting features

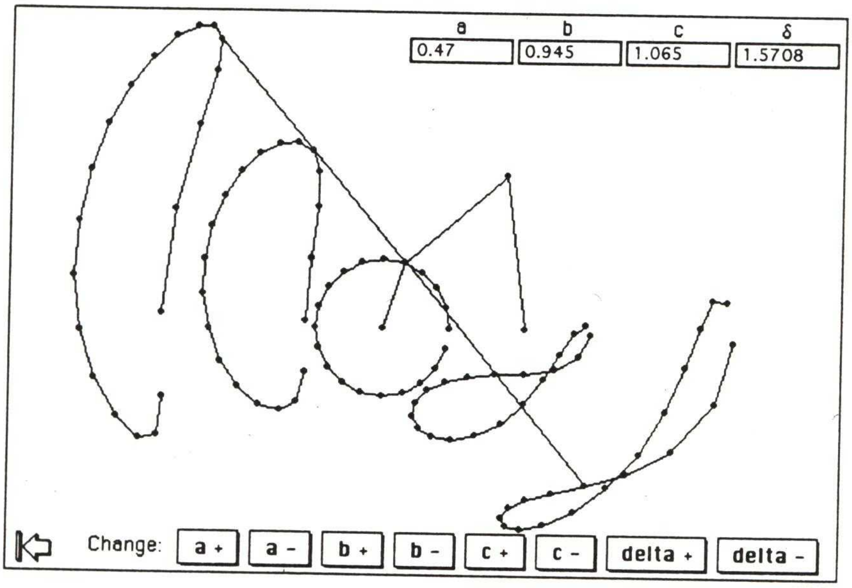


Fig. 13. Page of an atlas of mechanism trajectories.

of a standard computer simulation package (e.g. run velocity, 3D graphics, mathematical routines, optimization algorithms, etc.) are not provided by a typical hypertext. On the other hand, to develop them may turn out to be impractical. Therefore, it is useful to integrate the hypertext environment and other programming environments, also to focus the students' attention on more powerful and professional simulation methods. Such a possibility has been shown in Section 2 for a Pascal environment. Figure 14 presents the user's interactions with the mechanism model generator (in the hypertext environment) and the compiler (in the Pascal environment). In this way, the best features of both environments can be exploited.

4. REMARKS AND USERS' REACTIONS

We think that simplicity of use and consistency among different documents are of great importance for educational applications. Such features have been achieved by defining some basic 'styles' for managing the documents: opening, closing, and resuming actions have particular visual effects; only the data fields strictly necessary at a given stage are displayed and have context-dependent labels, and so on. Some styles have also been created to represent the active elements of the hypertext: any boldface piece of text corresponds to a navigation button, or starts an action described by the text itself; fields for input data are accessed by tabs or by mouse; fields for output data are locked, and so on. Moreover, the goal of maintaining similar graphic representations of the same object in different documents has been attained. Finally, typical features of computer-aided programs, including pull-down menu, file management, graphic animation, windows, and hard-copy capabilities, are provided.

As a hypertext system is open, it is an easy task to implement new document generators that make it possible to generate, in an automatic way, additional problem-oriented applications, using the existing simulators. For instance, the document containing an archive of mechanism designs can be obtained by managing various simulation runs and storing the corresponding results.

The proposed hypertext applications were used to support lectures for university courses in mechanical engineering. Students were very interested in the novel approach, which is very different from the simulation techniques commonly used in engineering education. Hypertext allows students to understand many theoretical and practical aspects of the problem they have to solve, and to perform personal investigations.

Students stated that they were able to understand the semantics and the information associated with mechanical problems through the interactions between browsing and simulation operations. They found the software implementation and the user interfaces very effective, and were particularly interested in the spatial organization of data and algorithms, in the asynchronous and non-linear interactions with the material, and in the structure of the text pages, which presented both a physical problem and its mathematical model. In our opinion, these are important remarks to be taken into account in developing educational software. The traditional simulation software is based on a temporal organization, as model design, model description, data input, simulation run, and data output are typical sequential operations. These tasks were usually accomplished through a batch process, and some important simulation packages for mechanism analysis still exhibit marked batch characteristics. On the other hand, the proposed hypertext approach is oriented toward a spatial distribution of objects, and the various simulation phases are interconnected and can be exploited under the user's complete control. Moreover, some kinematics applications that are traditionally presented on paper sheets (e.g. a mechanism atlas) are very difficult to use in an efficient and profitable way; on the other hand, their generation and non-

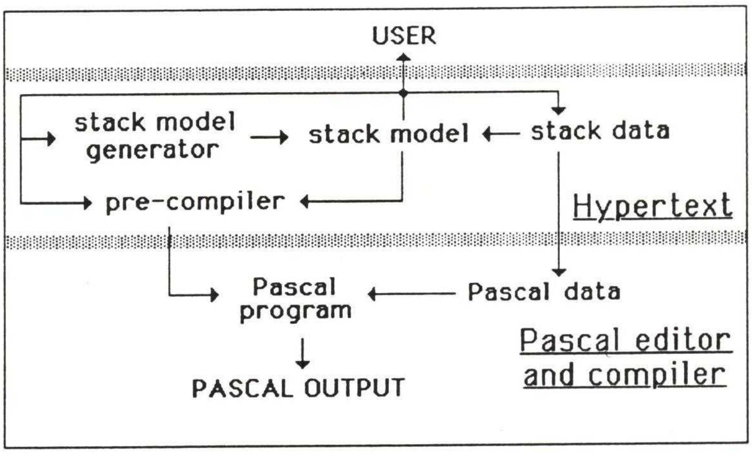


Fig. 14. Interactions hypertext/Pascal.

linear access by hypertext have been proved very interesting and have stimulated the students' creativity.

Clearly, the wide range of possibilities offered by the software allows students to work at different levels: they can choose to play a passive or active role, i.e. to use available material (e.g. a robot with a standard structure), or to create their own material (e.g. new robot structures). Moreover, they are completely free to change their role at any stage of their interaction with the hypertext system.

Finally, the hypertext environment also offers interesting possibilities to teachers to prepare the material for their lectures: it is easy to obtain transparencies related to theoretical topics, industrial applications, and specific numeric examples by exploiting the features of the hypertext system.

5. CONCLUSIONS

The stacks presented in the previous sections are examples of the potentialities of the proposed hypertext system for learning mechanical engineering. They can also be regarded as the 'cores' of more complex simulators and computer-aided tools that provide different interfaces and computational capabilities for specific classes of users.

Some of the salient hypertext features seem to be of particular interest for simulation purposes:

- the short time required to design and experiment software solutions oriented toward problems and users;
- the possibilities of fast prototyping and of analyzing alternative interfaces and software organizations;
- the spatial localization of data and algorithms, the asynchronous and nonlinear interactions with the material, and the possibility of structuring the cards in a way oriented towards both a physical problem and its mathematical model;
- the ease with which one can improve the performance of a stack and tailor them to specific requirements (for instance, outputs different from standard ones can be obtained by modifying some predefined cards or by inserting new ones, without any need for changing other parts of the stack);
- the opportunity of narrowing the gap between the software developer and users, stimulating the latter to become authors. Students can start by changing the aesthetic and presentation formats of a card, or they can add commentary cards without any programming training, and go through several authoring steps up to a complete self-developed application.

Further research work is currently in progress to interface the simulation stacks with other software packages, for instance, data bases and optimization programs. This kind of operation, which has proved useful for typical CAD programs, seems to be of great interest also for hypertext environments.

REFERENCES

1. K. Yamazaki and S. Miyazava, A development of courseware for mechatronics education, *Int. J. Appl. Engng Ed.*, **8**, 61–70 (1992).
2. R. F. Davey, Custom software for design education: an aeronautics example, *Int. J. Appl. Engng Ed.*, **5**, 383–389 (1989).
3. A. A. Seireg, Engineering education in the computer age, *Comp. Mech. Engng*, **2**, 2 (1984).
4. R. M. Akscyn, D. L. McCracken and E. A. Yoder, KMS: a distributed hypermedia system for managing knowledge in organizations, *Comm. ACM*, **31**, 820–835 (1988).
5. F. G. Halasz, Reflections on notecards: seven issues for the next generation of hypermedia systems, *Comm. ACM*, **31**, 836–852 (1988).
6. J. Savoy, Les sources des hypertextes: une bibliographie commentee, *Techn. Sci. Informat.*, **9**, 55–524 (1990).
7. P. F. Culverhouse, L. Ball and C. J. Burton, A tool for tracking engineering design in action, *Des. Stud.*, **13**, 54–70 (1992).
8. E. Giannotti and C. Galletti, Hypertexts for machine theory education, *Computers Ed.*, **16**, 121–126 (1991).
9. E. Giannotti and C. Galletti, Hypertext technology for simulation: impact on engineering curriculum, *Proc. 9th Int. Conf. on Technology and Education*, **1**, 164–167 (1992).
10. A. Hess, A CAI approach to teach human factors in engineering, *Int. J. Appl. Engng Ed.*, **7**, 358–367 (1991).
11. P. Fanghella, C. Galletti and E. Giannotti, Computer-aided modelling and simulation of mechanisms and manipulators, *Computer-Aided Des.*, **21**, 577–583 (1989).

Elena Giannotti received a degree in electronic engineering in 1973. She worked at ASW–SACLANT Center of La Spezia in 1973, as a researcher in acoustics. Then she received a research fellowship from the Ministry of Education to research in bio-engineering. From 1976 she lectured in Computer Sciences at the College of Engineering of the University of Genoa, centred on basics of Computer Sciences and Software Development. In 1982 she became Associate Professor of Computer Science at the Department of Communciations, Computer and Systems Sciences in the same University. From 1990 she has been Chairman of the

Council of Ingegneria Informatica at the University of Genoa. Elena Giannotti has worked in various fields of engineering: bio-engineering, computer-aided design, computer-aided instruction and visual programming. At present her primary interest is in educational applications of hypermedia, both for university regular curricula, and for industrial training.

Carlo Galletti, after taking a degree in mechanical engineering in 1969, worked at the Italian firm Zanussi. From 1973 he lectured in Mechanical Engineering, specializing in computer-applications and simulation, at the College of Engineering of the University of Genoa. In 1981 he became Professor of Mechanics of Machinery in the same University. For several years he was member of commission 'Linkages' of the International Federation of the Theory of Mechanisms and Machines. He has been consultant for training of various educational associations, such as IFOA of Reggio Emilia, and Consorzio Genova Formazione of Genoa. He is director of a national research project on Automation of Mechanical Design, involving 10 universities. His main fields of research activity are in engineering: mechanism theory, robotics, computer-aided design, and education. At the present time his primary interests are in robot kinematics, and in simulation for education.