Assur's-groups-based simulation for teaching kinematics of planar linkages

Carlo Galletti[1], Elena Giannotti[2]

[1]*Department of Mechanics and Design of Machines, University of Genoa, Italy*
E-mail: *carlo.galletti@unige.it*

[2] *Department of Communication, Computer and System Sciences, University of Genoa, Italy*

Keywords: Education, Linkage kinematics, Assur's groups, SIMULINK.

SUMMARY. In this work we present a simulation tool based on the method of Assur's groups to support teaching activities in the field of the kinematics of planar linkages. The tool is embedded in SIMULINK ad uses the block-oriented approach that is central both in SIMULINK modeling and in Assur's method. For each component useful for simulation according to Assur's method a SIMULINK masked block has been developed using subsystems and embedded MATLAB functions. Simple graphics is ready to use. Several examples are provided.

1 INTRODUCTION

The theory of Assur's groups - the first modular approach to the mechanism modeling - is a known method to describe, analyze, and synthesize linkages. It is based on observation that many mechanisms of technical interest can be build-up by selecting a small amount of arrangements of bodies and joints, connected in such a way that the kinematic equations of a whole mechanism can be formulated in a hierarchy of sub-systems and solved sequentially. The method has been extensively used for theoretical studies and for technical applications (a survey and extensive references on this subject are given in [1]).

Several software tools have been implemented in order to put into operation this approach. Moreover, its simplicity and intuitive characteristics make Assur's groups method suited for learning purposes [2, 3, 4]. However, existing implementations are not completely apt to this purpose. Often they are imbedded in very complex software with large capabilities (see, for instance [5]) that students must manage in order to use the software. In addition, the software implementation "hides" the underlying theory and the mathematical features of the method. This is a correct choice for professional activities, but is not fully satisfactory for learning purposes.

The following characteristics are important for a computer-assisted tool for education in linkage kinematics: 1) to be easy to use, 2) to present examples organized in advance, 3) to allow users to model a wide range of linkages of technical interest, 4) to furnish analysis results in numerical and graphic forms for plotting results and model animation. In our opinion three other characteristics can be useful: 5) to work in a simulation environment that the students already know, 6) to be allowed for exporting and importing data to and from this environment, 7) to let students to see the mathematics underlying the model simulation.

In order to satisfy all this issues, we present in this paper a simple simulation tool for performing kinematic analysis of planar linkages. An important point we want to outline is that we do not want that in order to use the tool students be obliged to learn a new software, but they can use a simulation approach that (hopefully) they have already learnt in their curriculum.

2 THE GROUPS

From a theoretical point of view the number of Assur's groups necessary to analyze any conceivable linkage is infinite, but very few groups are necessary to model a notable amount of mechanisms. For educational activities it seems profitable to have a certain amount in redundancy in the group choice, so that students can make better connection between a real mechanism and its model [6]. Therefore the groups adopted in our simulation are a few more than those strictly necessary.

Since the method of Assur's groups is naturally block-oriented, in order to develop the simulation tool three different types of kinematic blocks have been considered: non-active groups that are similar to the standard Assur's groups; active groups that realize kinematic drivers; special blocks that facilitate the modeling of linkages. The complete set of blocks is shown in Fig. 1.

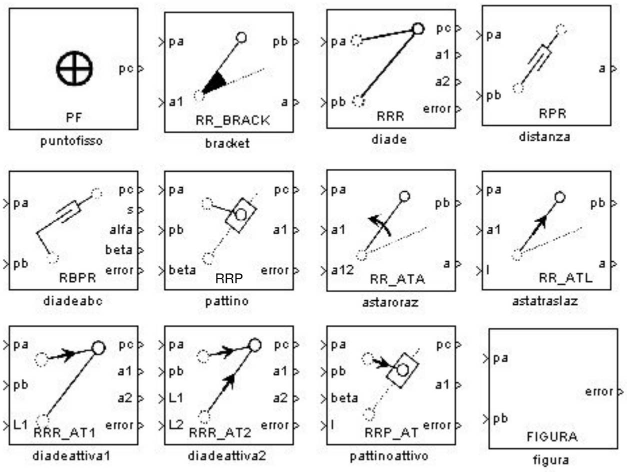


Figure 1: Kinematic blocks.

2.1 *Non-active groups*

Non-active groups correspond to link arrangements that do not change the mobility of any structure to which they are connected. They are labeled here according to the sequence of their joints, *RRR, RPR, RBPR, RRP* in Fig.1. All of them are connected to any external structure by revolute pairs, and one (*RRP*) requires also a connection by a sliding pair.

2.2 *Active groups*

Active groups have a structure similar to the non-active ones but provide one or two input motions. They are used to define the kinematic drivers of the linkage. In Fig. 1 they are *RR_ATA, RR_ATL, RRR_AT1, RRR_AT2, RRP_AT.* The first two blocks provide angular or linear motion, the third and the fifth can actuate the corresponding dyads, the fourth one can actuate a dyad RRR with two input motions.

2.3 *Special blocks*

The active and non-active groups provide all tools necessary for simulating a notable amount of linkages of practical interest. However, in order to build a linkage model in the easiest way and to display analysis results with more authentic appearance, three special blocks have been added, one for stating the positions of points fixed to a frame, a second one to extend the number of points and axes on a link, and a last one to display simple shapes of bodies. These are the blocks *PF, RR_BRACK, FIGURA* of Fig. 1. Some of their features are described in Section 5.

3 THE SOFTWARE DEVELOPMENT

As outlined in Introduction one of the purpose of this work is to provide a tool that does not require to learn a specific software, but only to use a general purpose simulation software in which the blocks of Fig. 1 can be easily manipulated. Since the Assur's method is characteristically block-oriented, a block-oriented simulation environment will exactly deal with the kinematic modeling. Therefore, we have adopted SIMULINK, a tool box of MATLAB, often used to support simulation in engineering curriculum.

Then, all blocks shown in Fig. 1 have been implemented as SIMULINK blocks and compose a kind of "menu" from which the blocks necessary for a specific simulation are selected (in fact Fig. 1 is a screen capture of this menu). This copy-and-paste procedure is exactly the same used to build any simulation model in SIMULINK and does not require any new understanding or programming.

When all necessary blocks have been pasted in a document, they can be connected according to the kinematic structure of the linkage. Also this operation is supported by SIMULINK block connections and the linkage model is ready to accept geometrical data.

In order to implement blocks in SIMULINK it is possible to use masked-blocks and to program MATLAB embedded functions. One masked block is provided for each block in Fig. 1. Under the mask of each block the connections of the block with input/output data are defined. For instance, Fig. 2 shows the connection provided under the mask of block RRR that models an RRR dyad.

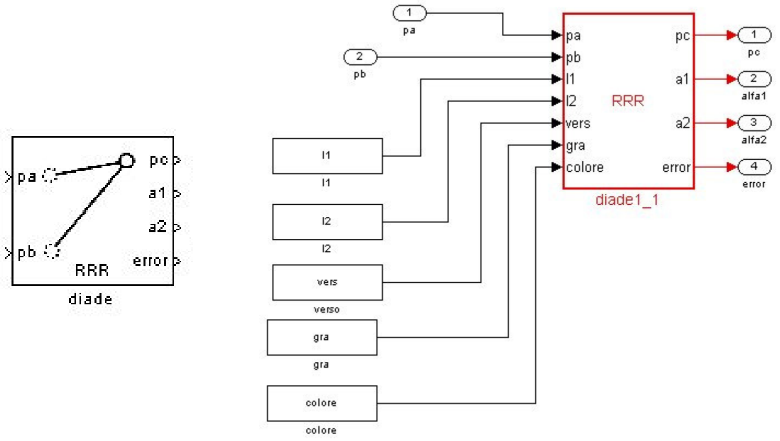
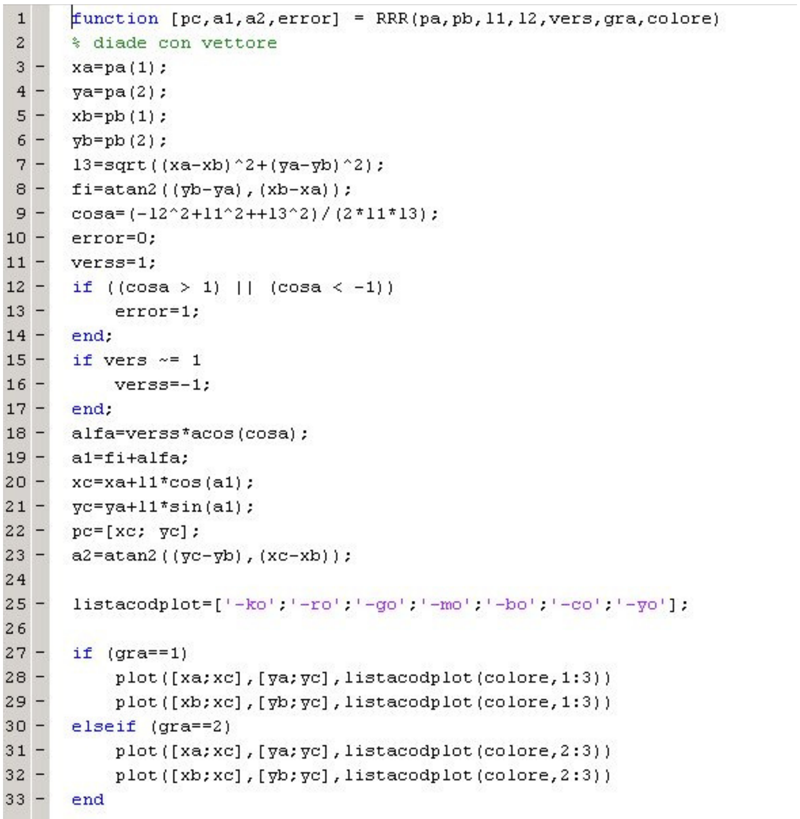


Figure 2: Block RRR under the mask

The left side of the figure shows that this block receives 2 input variables (1) and (2) from other blocks: these are the points pa and pb that define the position of the two external revolute joints of the dyad. Moreover, 5 constant parameters will be provided before the beginning of simulation. They are the lengths *l1* and *l2* of the links of the dyad and a flag *verso* used to select between the different branch of positions of the dyad. Two more parameters *gra* and *colore* furnish information about the graphic rendering of the dyad and will be described below.

The right side of the figure shows that the block provides 4 output variables to be used by other blocks and/or to be viewed by the user of the simulation: (1) the position *pc* of the intermediate point of the dyad, (2) and (3) the angles *a1* and *a2* formed by the links of the dyad with the x axis of a fixed reference, and an error flag, *error*, which is set by the dyad algorithm if the dyad cannot cope with the linkage constraints.

The programming of the dyad algorithm has been implemented by embedding a MATLAB function in the block RRR in the middle of Fig. 2. Fig. 3 shows the program instructions.



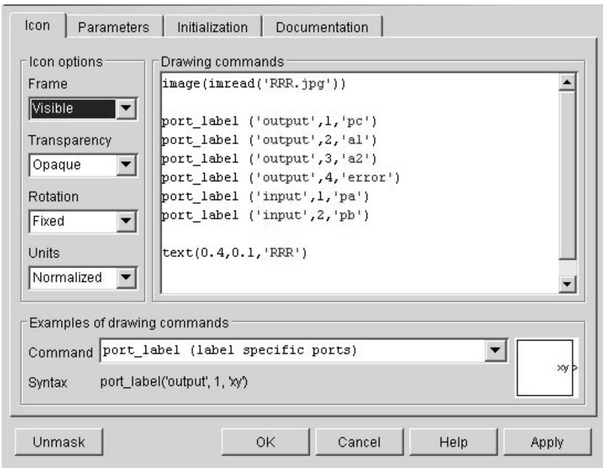
```
 1     function [pc,a1,a2,error] = RRR(pa,pb,l1,l2,vers,gra,colore)
 2     % diade con vettore
 3 -   xa=pa(1);
 4 -   ya=pa(2);
 5 -   xb=pb(1);
 6 -   yb=pb(2);
 7 -   l3=sqrt((xa-xb)^2+(ya-yb)^2);
 8 -   fi=atan2((yb-ya),(xb-xa));
 9 -   cosa=(-l2^2+l1^2++l3^2)/(2*l1*l3);
10 -   error=0;
11 -   verss=1;
12 -   if ((cosa > 1) || (cosa < -1))
13 -       error=1;
14 -   end;
15 -   if vers ~= 1
16 -       verss=-1;
17 -   end;
18 -   alfa=verss*acos(cosa);
19 -   a1=fi+alfa;
20 -   xc=xa+l1*cos(a1);
21 -   yc=ya+l1*sin(a1);
22 -   pc=[xc; yc];
23 -   a2=atan2((yc-yb),(xc-xb));
24
25 -   listacodplot=['-ko';'-ro';'-go';'-mo';'-bo';'-co';'-yo'];
26
27 -   if (gra==1)
28 -       plot([xa;xc],[ya;yc],listacodplot(colore,1:3))
29 -       plot([xb;xc],[yb;yc],listacodplot(colore,1:3))
30 -   elseif (gra==2)
31 -       plot([xa;xc],[ya;yc],listacodplot(colore,2:3))
32 -       plot([xb;xc],[yb;yc],listacodplot(colore,2:3))
33 -   end
```

Figure 3: Program of RRR embedded function

Instructions from 1 to 23 carry out the closed form position analysis of the dyad, using the input variables *pa, pb, l1, l2, vers* listed in row 1. The remaining instructions are used to obtain a specific graphic display of the links (see Section 5).

4 PARAMETERS DEFINITION

Using the SIMULINK block programming we have developed an user-friendly interface for students that allows them to input all data for each block. The mask of each block has been designed to represent the geometrical shape of the corresponding group; input and output are provided and labeled for connecting the various blocks of the model (Fig. 4).



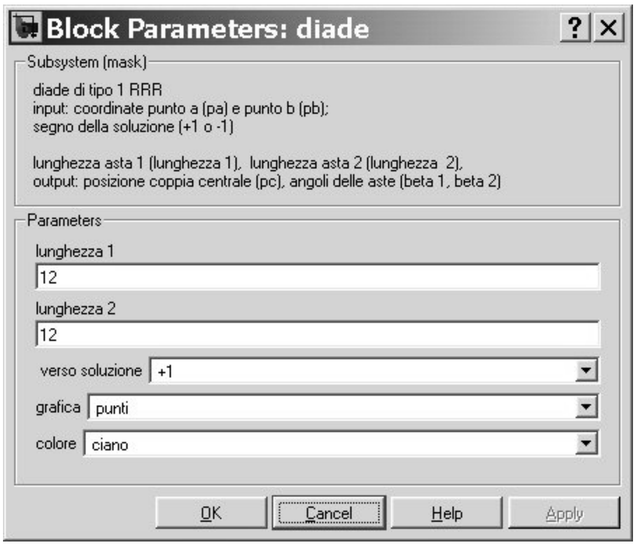
Frame: Visible
Transparency: Opaque
Rotation: Fixed
Units: Normalized

```
image(imread('RRR.jpg'))

port_label ('output',1,'pc')
port_label ('output',2,'al')
port_label ('output',3,'a2')
port_label ('output',4,'error')
port_label ('input',1,'pa')
port_label ('input',2,'pb')

text(0.4,0.1,'RRR')
```

Command: port_label (label specific ports)
Syntax: port_label('output', 1, 'xy')

Figure 4: Design of the mask of block RRR

A double click on each block opens a dialog box for entering block parameters, consistent with the SIMULINK standard. Figure 5 shows the dialog box of the block RRR.



diade di tipo 1 RRR
input: coordinate punto a (pa) e punto b (pb);
segno della soluzione (+1 o -1)

lunghezza asta 1 (lunghezza 1), lunghezza asta 2 (lunghezza 2),
output: posizione coppia centrale (pc), angoli delle aste (beta 1, beta 2)

lunghezza 1: 12
lunghezza 2: 12
verso soluzione: +1
grafica: punti
colore: ciano

Figure 5: User's interface of block RRR

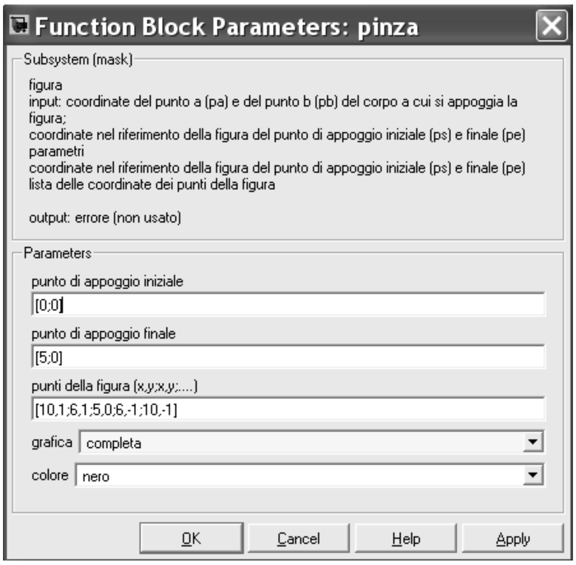
The window provides some information about the variables and the parameters of the block at the top, fields for entering the values of the parameters, and popup fields to specify graphic options. In this case the user must give the lengths of the two links and a parameter (*verso della soluzione*) for selecting between the two solutions of the position equations. Two other set of parameters can be given in order to drive the graphic rendering of the linkage and will be described in the following section.

By clicking on the button *Help* in the bottom of the window, a new window with information specific of the RRR dyad will pop up. Here it is possible to insert learning material about the group, as its closure equations, solution algorithms, condition of solution existence, etc., supporting the student's understanding of the topic.

5 GRAPHIC RENDERING

The blocks of the standard *sink* menu of SIMULINK can be used to display simulation results (e.g., *Scope*, *XY Graph*, *To Workspace*). However, in mechanism analysis some type of animation of the mechanism is mandatory for understanding its motion. Therefore, the possibility of drawing a simple but expressive draft of the kinematic chain described by the model has been provided. Each block is programmed in such a way that a line corresponding to a link and/or a circle corresponding to the points of the block can be sketched in a figure. The programming is very easy: for instance, the instructions to sketch the two links and the three points of a dyad RRR are shown in lines 25-32 of Fig. 3.

Finally, a further feature has been provided in order to make it possible to draw more complex body shapes. To this purpose, a special block *FIGURA* has been implemented (last block in Fig. 1) that permit to define a shape that the software uses to apply it to any two points of the mechanism (Fig. 6).



Function Block Parameters: pinza

Subsystem (mask)

figura
input: coordinate del punto a (pa) e del punto b (pb) del corpo a cui si appoggia la figura;
coordinate nel riferimento della figura del punto di appoggio iniziale (ps) e finale (pe)
parametri
coordinate nel riferimento della figura del punto di appoggio iniziale (ps) e finale (pe)
lista delle coordinate dei punti della figura

output: errore (non usato)

Parameters

punto di appoggio iniziale
[0;0]

punto di appoggio finale
[5;0]

punti della figura (x,y;x,y;....)
[10,1;6,1;5,0;6,-1;10,-1]

grafica: completa

colore: nero

Figure 6: Input of a simple shape

6 EXAMPLE

Figure 7 shows an anthropomorphic robot arm with its actuating kinematic chain. A scheme of the planar chain that drives the second and the third joint of the arm (joints A and H) is shown in the same figure. This planar chain is used as an example of the mechanism description and analysis, choosing link 1 as a frame. The resulting SIMULINK block diagram is shown in Fig 8. The blocks define: 4 fixed points A, B, D, E, 2 active dyads DGA, EFB (providing the drivers), one dyad CLH, and 3 brackets AH, BC, LP to model the corresponding links. The blocks are connected according to the kinematic structure of the mechanism.

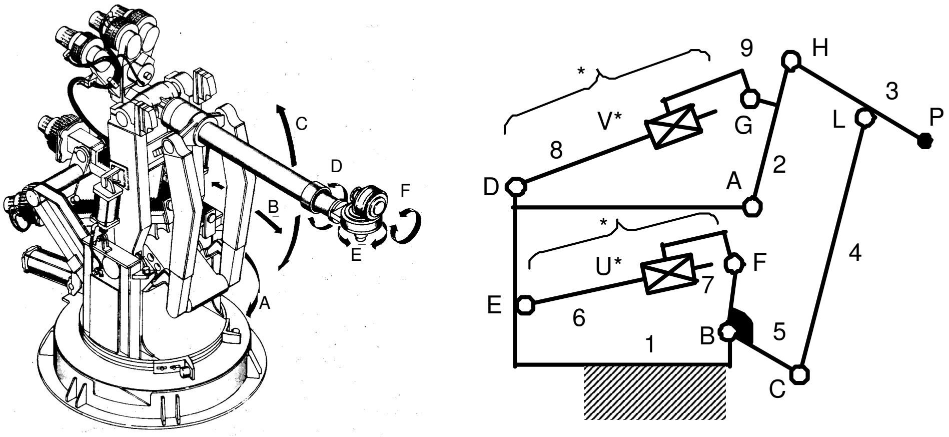


Figure 7: An anthropomorphic robot and its kinematic chain (simplified)

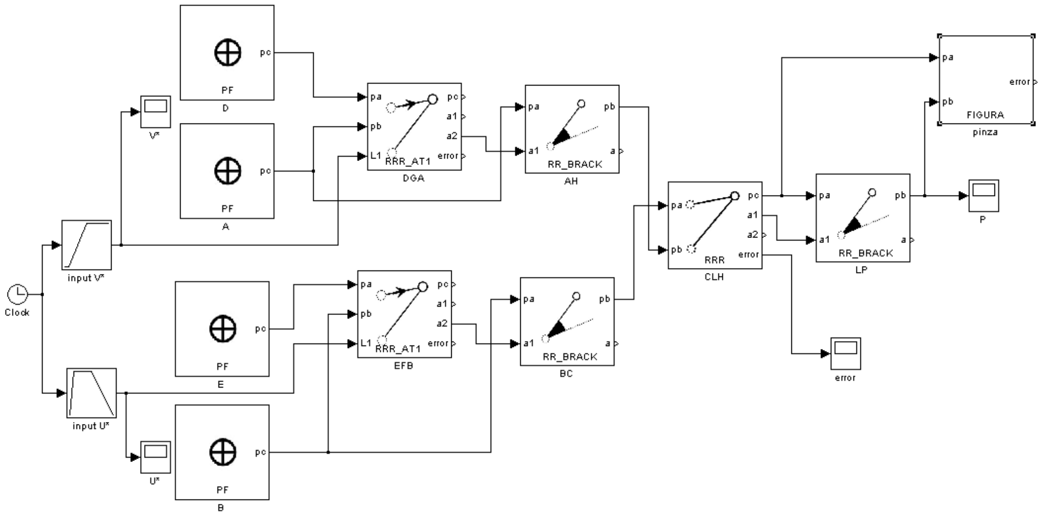


Figure 8: Kinematic model of the chain of Fig. 7

The laws of motion of the drivers (the distances DG and EF, i.e., the lengths of the active links of the active dyads) are given by two standard blocks *Lookup Table.* whose input is "time". A block *FIGURA* has been applied to points L and P of the scheme in Fig. 7 to represent an end-effector with two fingers.

Curves containing output variables of the simulation are displayed by several scopes - in Fig. 8 there are 4 scopes: P for the output of end-effector point, V* and U* for monitoring the driver input data, *error* to monitor a potential error in dyad compatibility. Displays of the scopes reporting the x and y coordinates of point P, and the value of *error* are shown in Fig. 9. The scope of error (always equal to zero) shows that no errors in closure equation have been detected during the simulation.

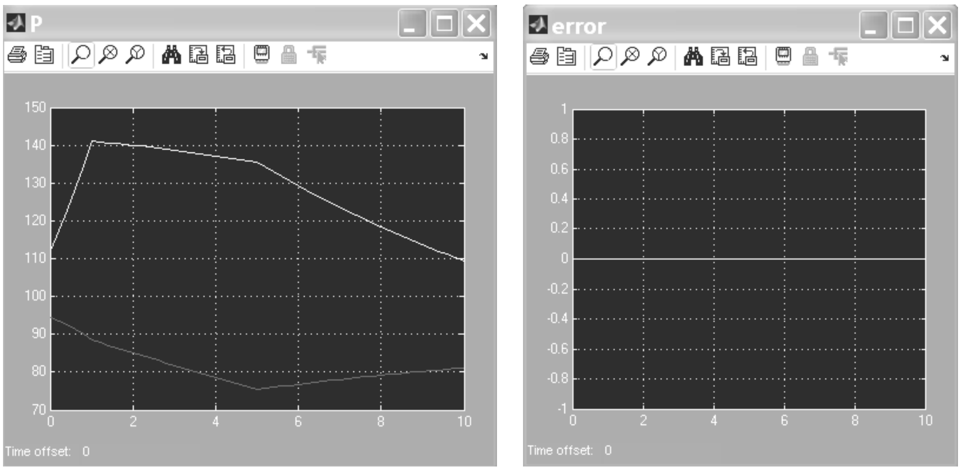


Figure 9: Scope output

Moreover, the position of each link can be drawn. Fig. 10 shows several drawing alternatives that can be selected at run time using popup menus for each block like the two shown at the bottom of Fig. 5: the chain in its initial position, the chain positions according to the motion of the drivers, the positions of the 2 active dyads and the trajectory of point P of the end-effector, and the drawing of the two fingers of the end-effector all through the motion.

7 STUDENT INTERACTIONS

Using SIMULINK and the blocks we have described in previous sections, several possibilities for supporting learning of mechanism kinematics can be devised.

1- a teacher can prepare models of many linkages implemented in SIMULINK and give students them as "canned" systems: students can run simulations with these models, modify parameters and visualize the resulting performances of the mechanisms.

2- a teacher can give students kinematic schemes of mechanisms from which students can derive the Assur's modeling and develop a SIMULINK model by which they are able to study the mechanism performances.

3- results of simulation can be exported to MATLAB environment for any mathematical operation combining the simulation results, e.g., to obtain numerical derivatives of the positions or to compare the effects of different set of parameters with respect to a given optimization criterion.

4- from MATLAB environment results can also be exported to different programs, as EXCEL, for further computations and rendering. Moreover, simulation results and graphics can be used to prepare report on laboratory activities.

5- if students are required to gain knowledge of algorithms of kinematic analysis, a "look under the mask" of the block can support such learning (Fig. 2 and Fig. 3). Moreover, the *Help* button provided in the dialog boxes of each block links to any material concerning the group properties, algorithms, and so on that a teacher want to display in the learning phases.

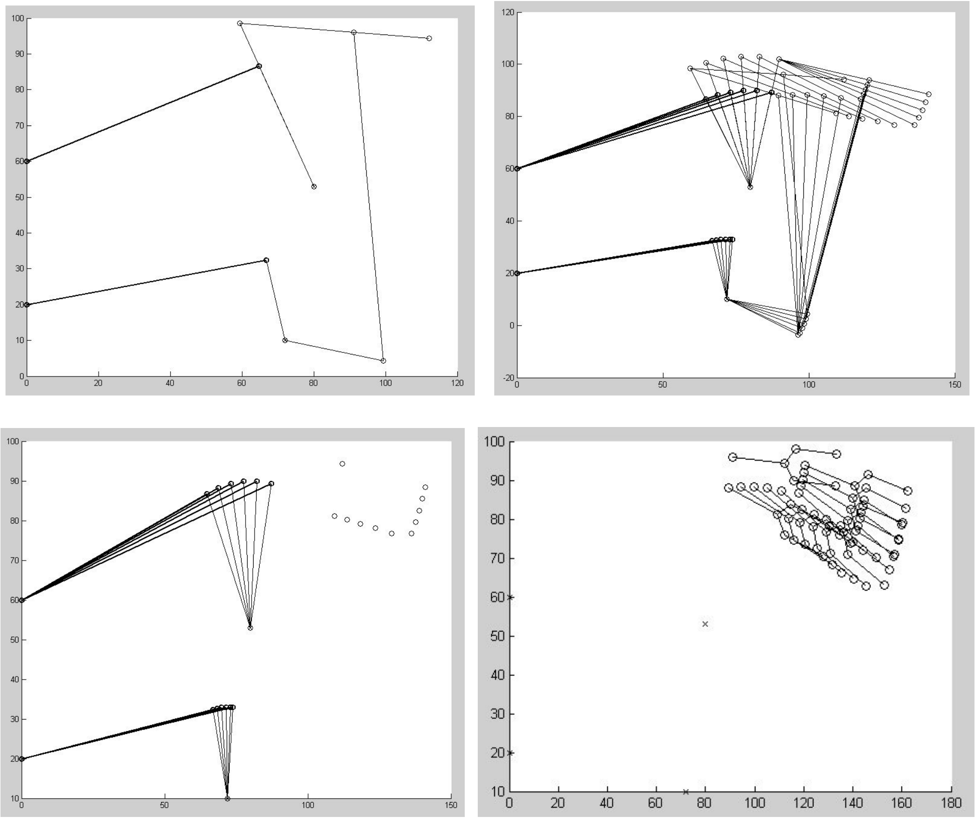


Figure 10: Graphic output of model of Fig. 7

8 CONCLUSIONS

The software tool we have presented in this work offers several features useful for supporting learning of linkage kinematics. Its main characteristics are:

- it is based on the Assur's group method for modeling and solving closure equations.
- users are not required to do any explicit programming and do not need to learn a specific analysis program.
- the computer modeling is based on SIMULINK, which is often a simulation tool that students know.
- several interactions are possible with the software: to use only models prepared in advance, to develop new models, to see how kinematic analysis algorithms work.
- simulation results can be immediately exported to the MATLAB environment using *To*

Workspace blocks and to other external programs for further result manipulations that can complement the simulation.

Based on these capabilities we think that this tool can be very useful to support student training in kinematics.

References

[1] Ceresole, E., Fanghella, P. and Galletti, C., “Assur's Groups, AKCS, Basic Trusses, SOCS, etc.: Modular Kinematics of Planar Linkages,” in *Proc. 1996 ASME DETC & CIE Conferences,* 96-DETC/MECH-1027 (2002).

[2] Cossalter, V., *Meccanica applicata allel macchine*. Edizioni Progetto, Padova (2002).

[3] Giannotti, E, Fanghella, P. and Galletti, C., “Hypertexts for Machine Theory Education,” *Computers Educ.*, 16, 1, 121-126 (1991).

[4] Tang, L. and Sun, X-Q, “Method and Realization of Computer-Aided Combination of Assur Groups in Conceptual Design of Planar Linkage Mechanisms,” in *Reconfigurable Mechanisms and Robots,* (Dai, J. S., Zoppi, M. and Kong X., Eds.), KC Edizioni, ISBN 978-88-89007-37-2, 123-128 (2009).

[5] Kecskemethy, A., “Oject-Oriented Modelling of mechanical Systems,” in *Kinematics and dynamics of Multi-Body Systrems,* (Angeles, J. and Kecskemethy, A., Eds.), Springer Verlag, 217-276 (1995).

[6] Giannotti, E, and Galletti, C., “An Approach Based on Simulation and Hypermedia Technology to a New Courseware in Mechanics of machines,” *European Journal of Engineering education*, 21, 3 (1996).