CONAT20044090

AUTOMATION TOOL FOR THE 2D COCKPIT ARCHITECTURE RESEARCH

[1]Şişman, Viorel, [1]Covaciu, Dinu

[1]Transilvania University Brasov, Romania

KEYWORDS

CAD, automation tool, cockpit architecture, mannequin, algorithm

ABSTRACT

The ideal solution to help ensure that the finished (car) product meets the ergonomics requirements is to conduct the traditional mock-up stage using computer graphics. Starting from the working principles, the algorithm and the anthropometrical 2D CAD model presented in the previous study ("Cockpit Architecture Research Using The New 2D Mannequin Concept"), the authors conceived a new software for the 2D cockpit architecture research.

This paper presents a software automation tool developed using the AutoLISP language, which runs under the AutoCAD system. The program uses extended AutoLISP functions to add custom data to the drawing objects, and use this data to manipulate mannequins with different sizes.

The program is very useful in the early design and styling stage.

MAIN SECTION

For each alternative of a new body design proposed by the stylist it is necessary to make an ergonomic study of the cockpit. The proposed software named “MANE” simplifies the process and makes it more rapid.

As a case study it was proposed a new car concept that is shorter and higher than a classical concept of a city car. The new concept is conceived for the transport of a greater number of passengers in a more limited space of the cockpit. In the figure 1 it was proposed the new car profile in two alternatives - having the same proportions but different dimensions. One of the most important difference between the two cars is the interior height.

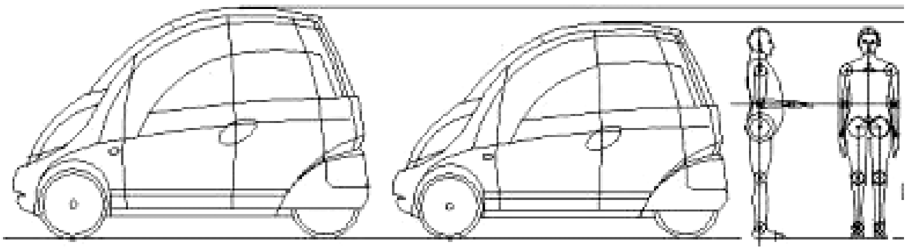


Fig. 1

First, it was analyzed the smaller automobile.

The reference point of the mannequin is the heel point leaned on the floor. The analyze is made for the two extreme mannequins usually used in the ergonomics – the 95%male percentile and the 5%female percentile. There are imposed angular restrictions for the mannequin joints (figure 2).

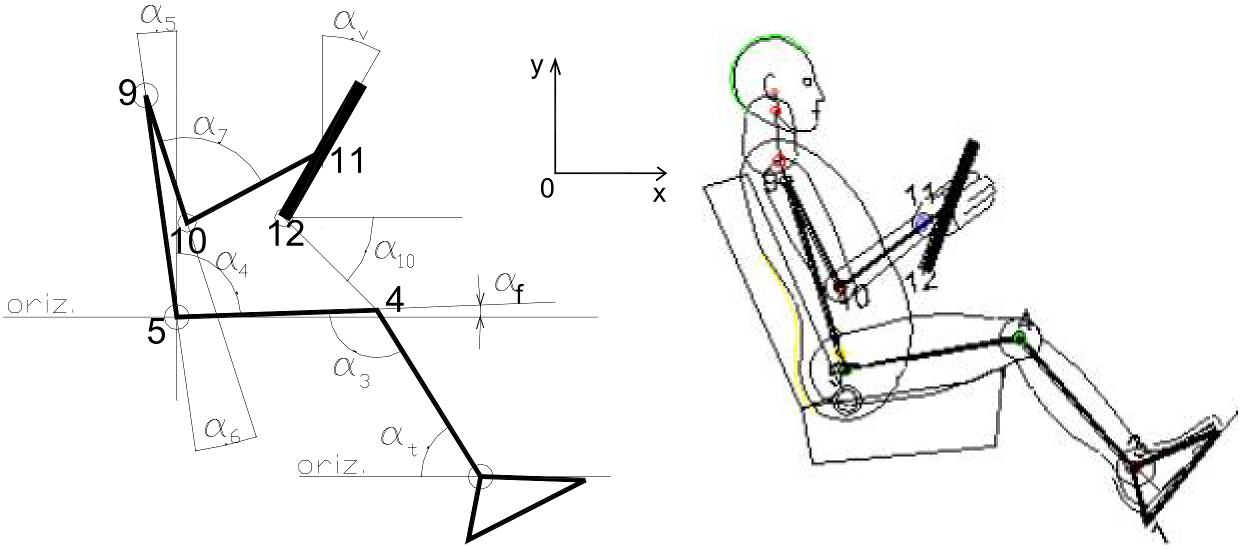
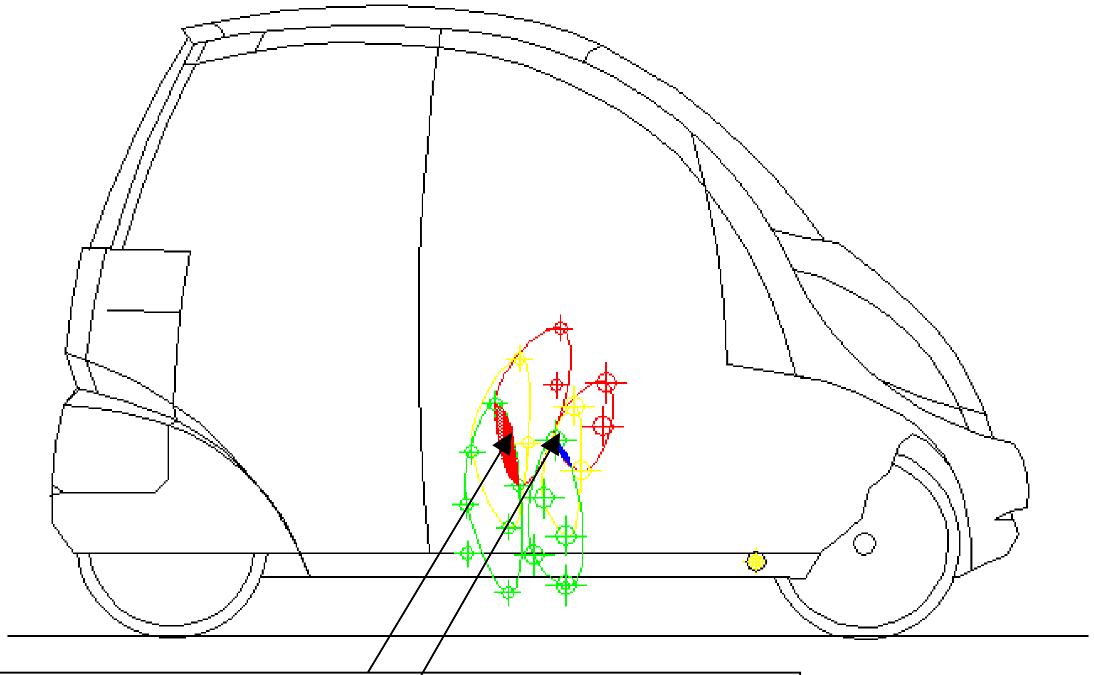


Fig. 2

The program "MANE" determines the hip domain (of the point 5) for the two mannequıns (figure 3).



The hip domain (of the point 5) for the 95% male percentile

The hip domain (of the point 5) for the 5%female percentile mannequin

Fig. 3

It is important to notice (from the figure 3) that for each mannequin there is an intersection of the hip point domains corresponding to each position of the acceleration pedal. In the example it is assumed that the angle α_1 takes three values and for each value it is represented a distinct domain of the point 5. The intersection sub-domain of the points 5 provides the comfort for the mannequin legs for all the acceleration's positions assumed.

From the two intersection domains there are chosen two points, one for each mannequin. The "MANE" program automatically represents the two mannequins profile (figure 4).

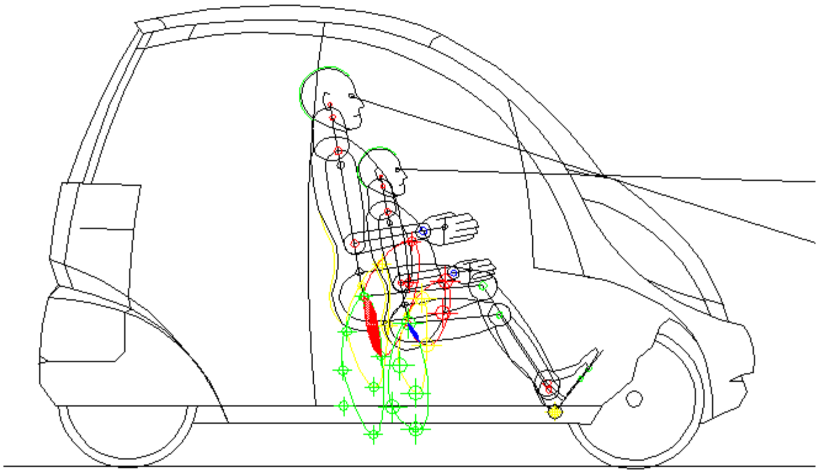


Fig. 4

The gap between the two points include any possible seat adjustment.

It can be observed that the vision field of the 5% female percentile mannequin is limited below the recommendations (figure 5).

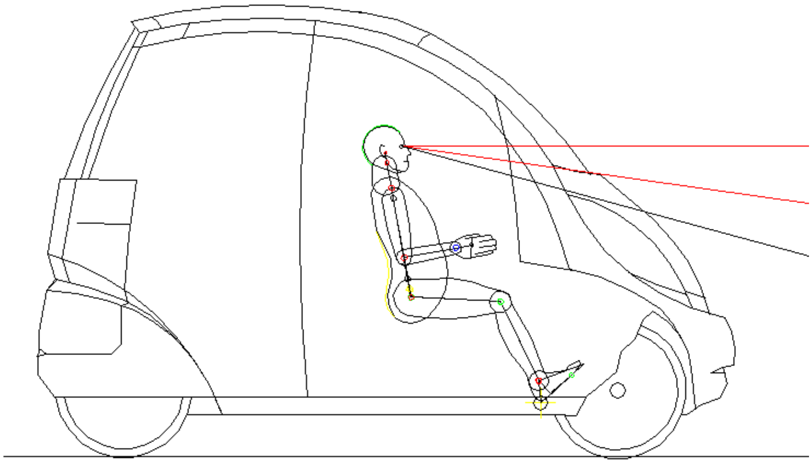


Fig. 5

As a consequence, the lower limit of the windscreen is modified (figure 6).

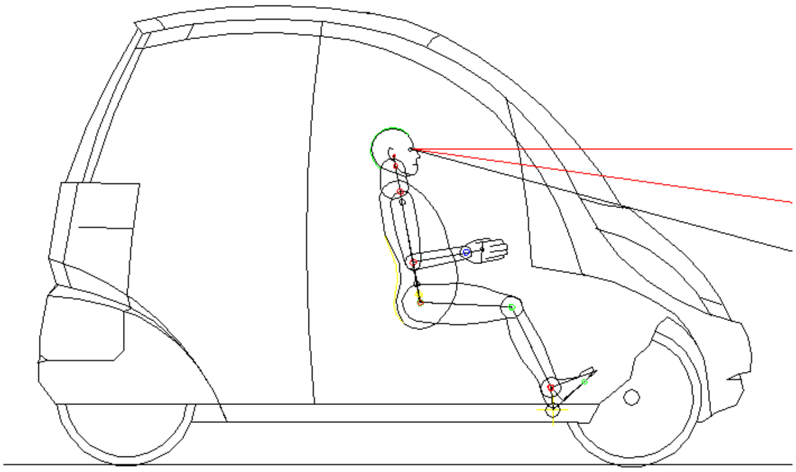


Fig. 6

In this way the ergonomic study influence the stylistic concept.

By choosing the type of the mannequin and the corresponding hip point 5, the “MANE” program determine and represents the comfortable positions of the steering wheel 12. The positions also depend on the diameter and the inclination of the steering wheel (figure 7). It is important to notice that it is preferable to find a common sub domain of the points 12 for a given inclination angle and a common diameter of the steering wheel for the two mannequins used and for all possible users. In this sub domain it can be placed the 12 point of a fixed steering wheel. It was noticed that generally the steering wheel must be adjustable. The “MANE” program gives the representations of the possible domains of the steering wheel that provide the comfortable conditions for the arms.

In the figure 8 it is represented one of the used mannequins placed in the driving position and the determined domains of the points 5 and 12.

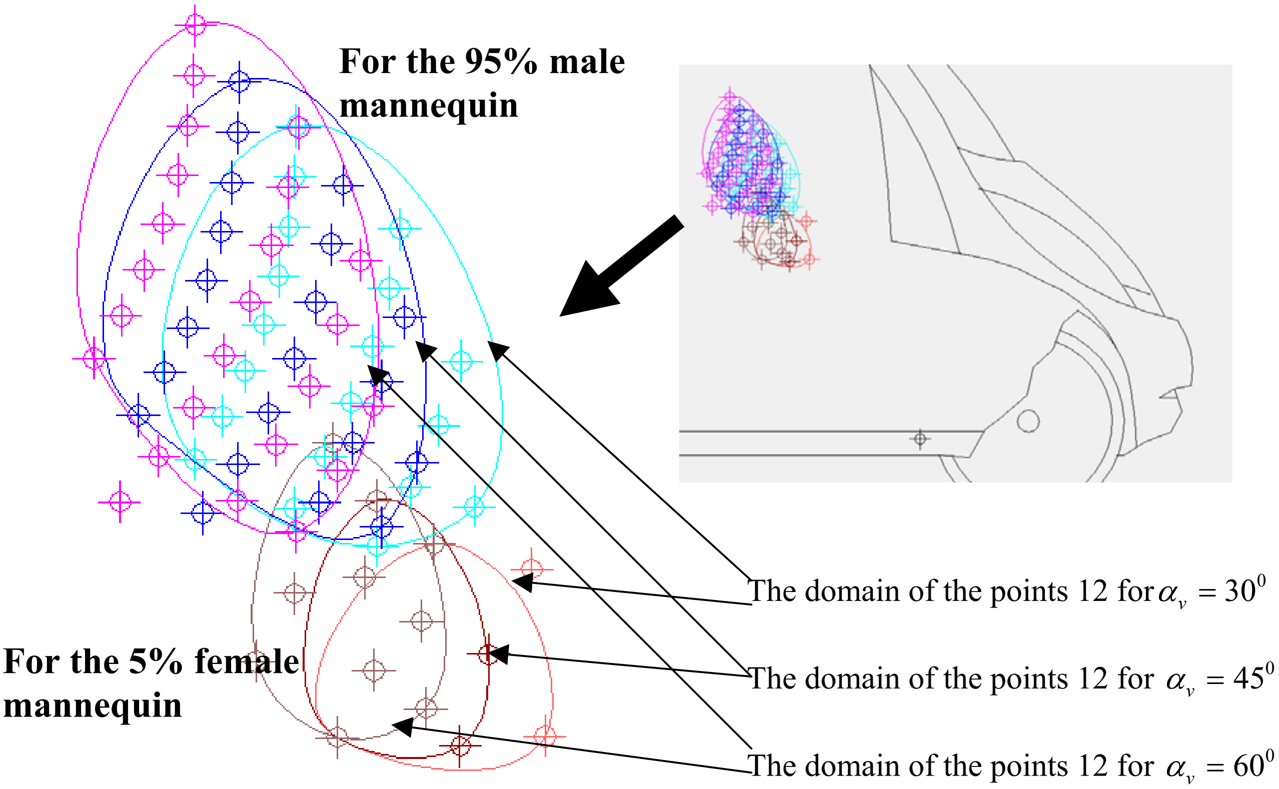
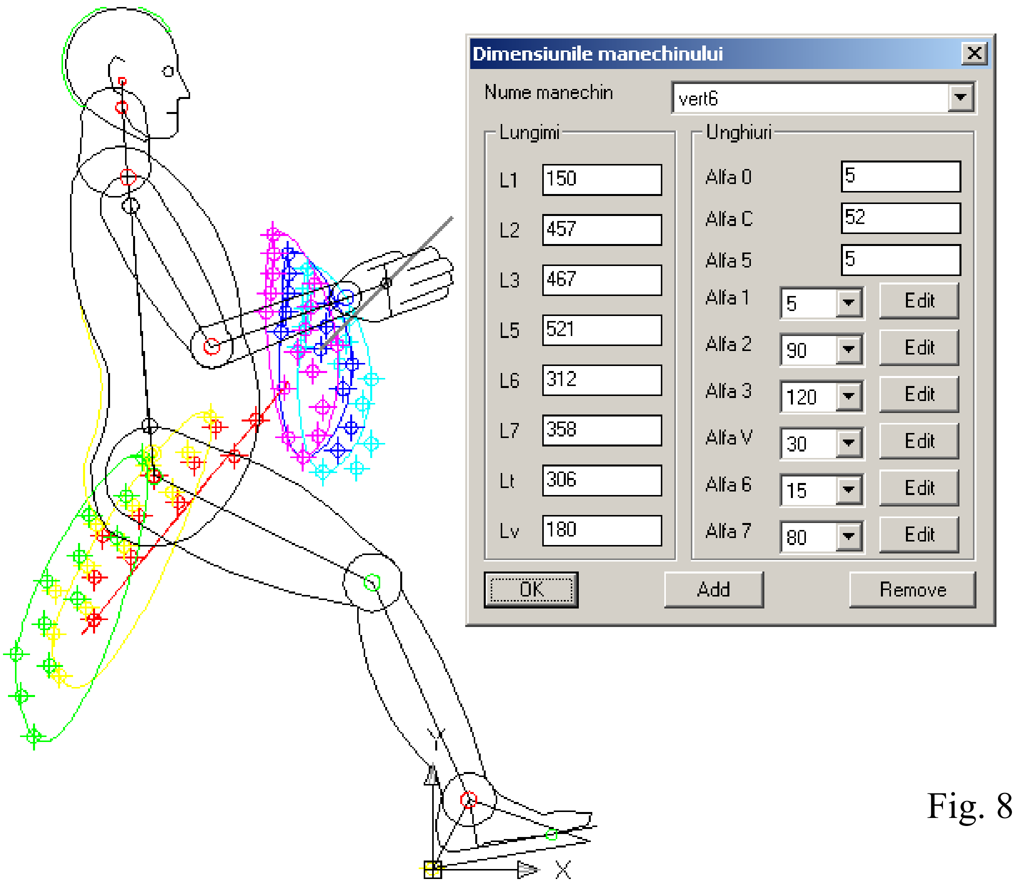

For the 95% male mannequin
For the 5% female mannequin
The domain of the points 12 for $\alpha_v = 30^0$
The domain of the points 12 for $\alpha_v = 45^0$
The domain of the points 12 for $\alpha_v = 60^0$

Fig. 7



Dimensiunile manechinului

Nume manechin: vert6

Lungimi

L1	150
L2	457
L3	467
L5	521
L6	312
L7	358
Lt	306
Lv	180

Unghiuri

Alfa 0	5
Alfa C	52
Alfa 5	5
Alfa 1	5
Alfa 2	90
Alfa 3	120
Alfa V	30
Alfa 6	15
Alfa 7	80

Fig. 8

The figure 9 represents the two mannequins (95%male percentile and 5%female percentile) in the driving positions determined with the program “MANE”. It is possible to approximate the domain of vision using the eyes of the represented mannequins as a point of view (figure 10).

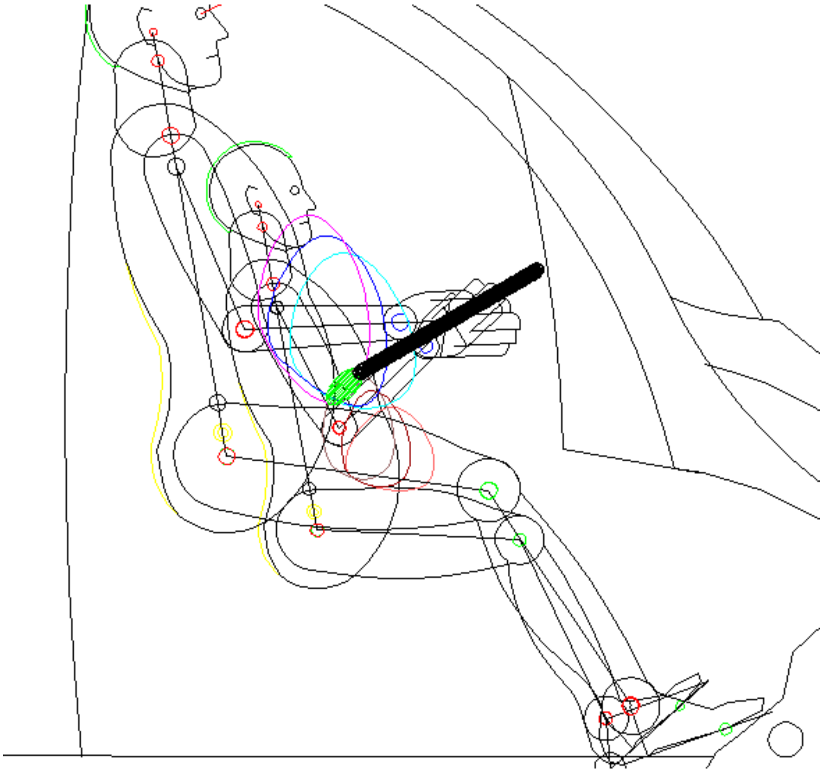


Fig. 9

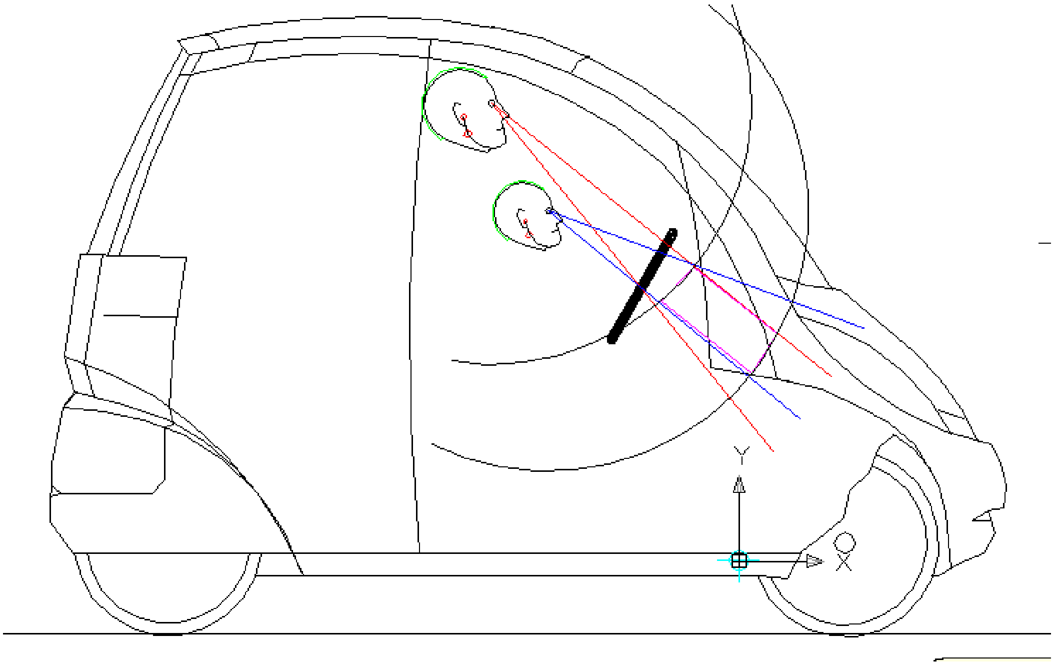


Fig. 10

The architecture of the cockpit is complete with the panel work plane placed in a domain that provides a very good vision for the drivers (figure 11).

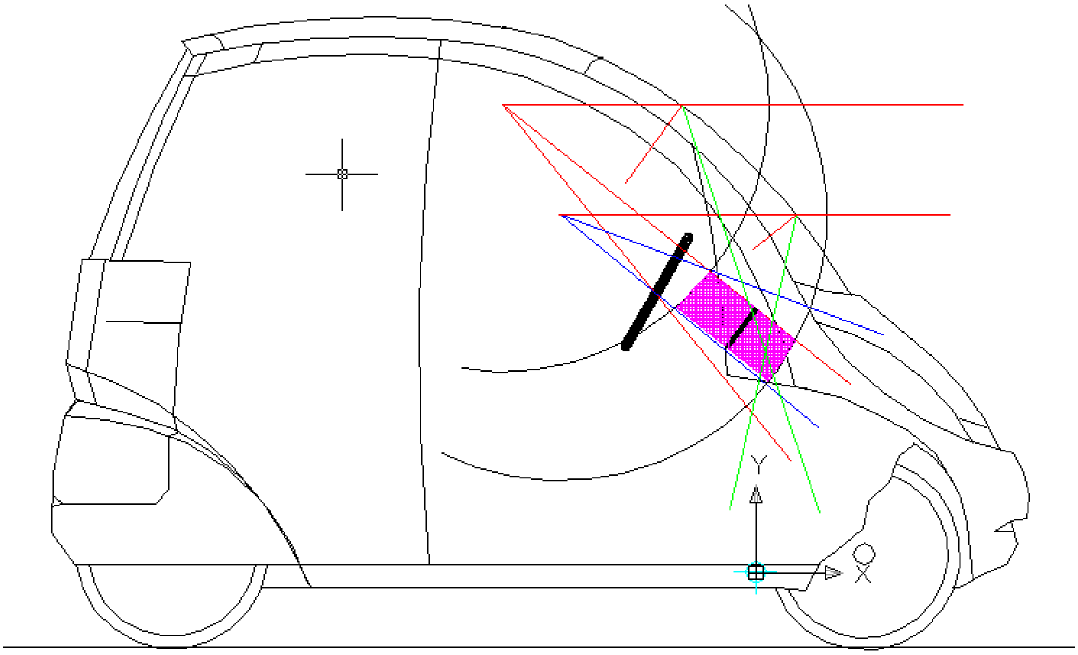


Fig.11

In the figure 12 it is represented the basic architecture of the cockpit.

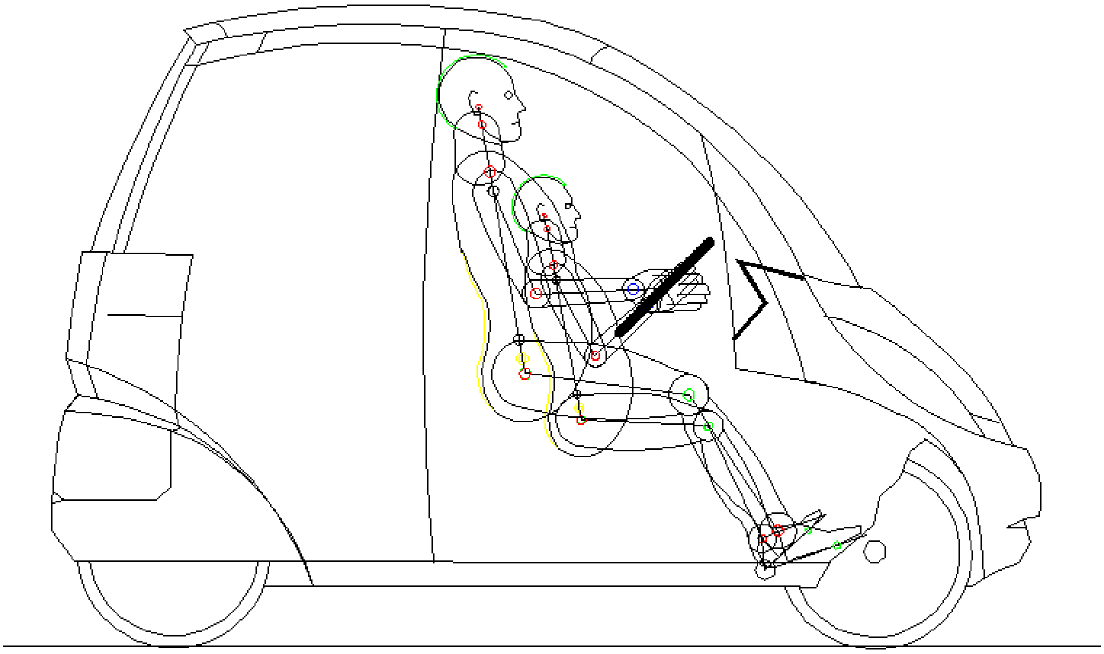


Fig. 12

Using the same methodology it was analyzed the bigger version of the new car concept. The “MANE” program allow the study of very different ergonomic solutions, including the possibility of a vertical travel position of the passengers that permits an important decrease of the automobile length or an important increase of the transport capacity (figure 13 and figure 14).

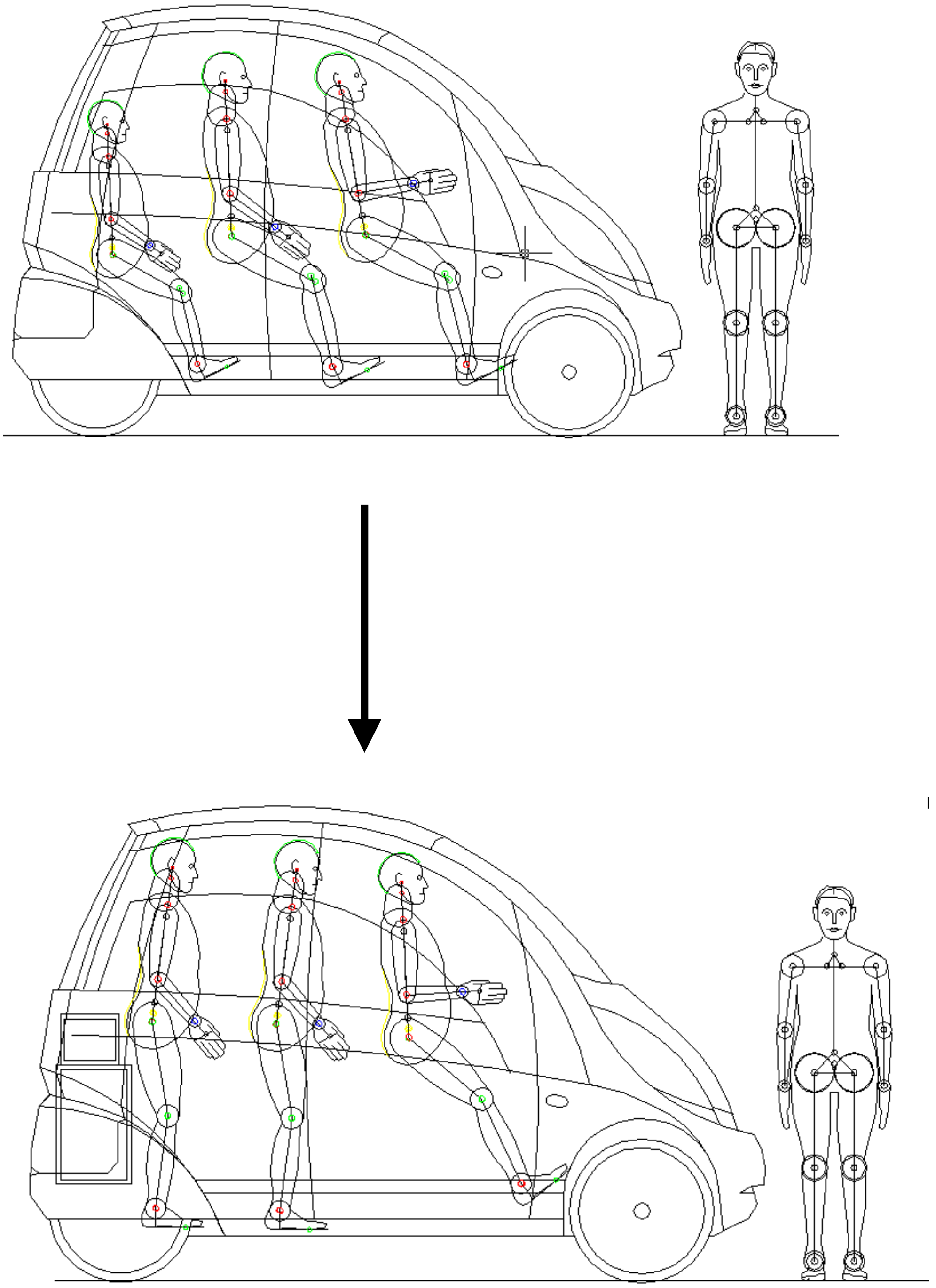


Fig. 13

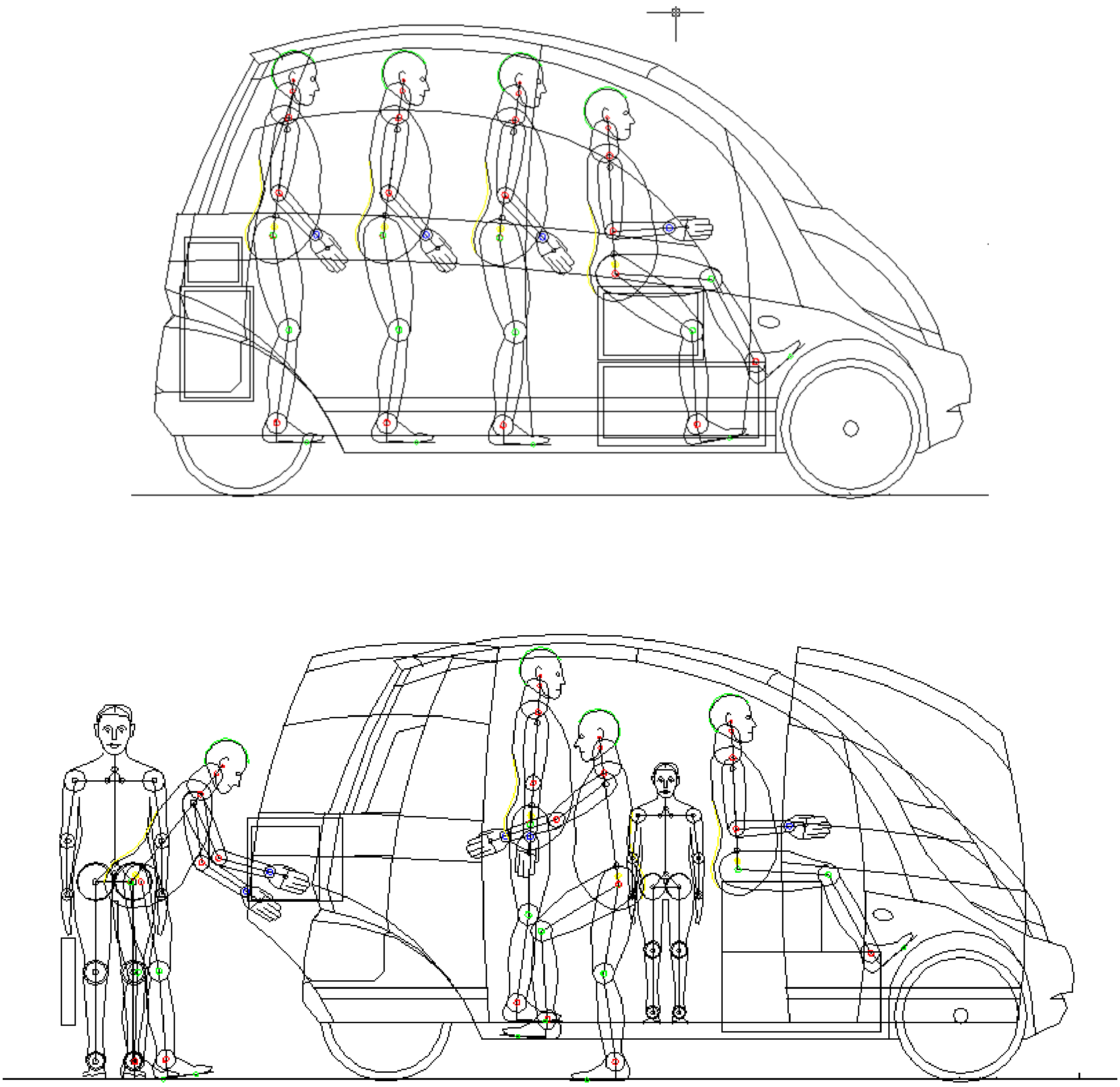


Fig. 14

Interesting conclusions can be drawn regarding the driver position and the utilization of the cockpit space.

The “MANE” program proven to be a very useful instrument for the body car design because of the possibility to simulate many different ergonomic conditions.

Now let’s take a look inside the software. This is written using the AutoLISP language, which is for years the standard for customizing AutoCAD. Now Visual LISP (VLISP) represents the next generation of LISP for AutoCAD, by adding significantly more capabilities. Our application uses some these capabilities.

In order to allow access to the input data for each case study, we added custom data to the drawing objects. An older way is using extended data (xdata), which is created by applications written with ObjectARX or AutoLISP.

We chosen for this application another way: using the interface with objects via the Microsoft ActiveX Automation interface. First, the AutoCAD entities was transformed in VLA objects: `(vlax-ename->vla-object entname)`.

A custom defined function was used to attach data to each object. An example code is below:

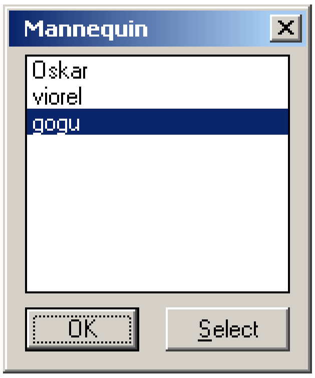
```
;;put data to a given entity
(defun put_data(e key data)
;;e is the entity; key identifies the data store (more stores can be
;;set for each object)
;;data is a lisp list containing custom data
  (vlax-ldata-put e key data)
  )
```

In the same way, the stored data can be read from the existing objects:

```
;;get data from a given entity
(defun get_data(e key)
;;return a lisp list containing custom data
  (vlax-ldata-get e key)
  )
```

The data attached to the entities (VLA objects) refers to the input data: lengths, angles, mannequin name. These info helps the user to easily identify each case studied when more was taken into consideration. Frequently is necessary to display only one of the mannequins and hide all the others. This is possible using a simple dialog box containing a list with the names of all defined mannequins. The mannequins have different layers, to easily hide/display them and their auxiliary entities (input points, point area contours, skeleton and body).

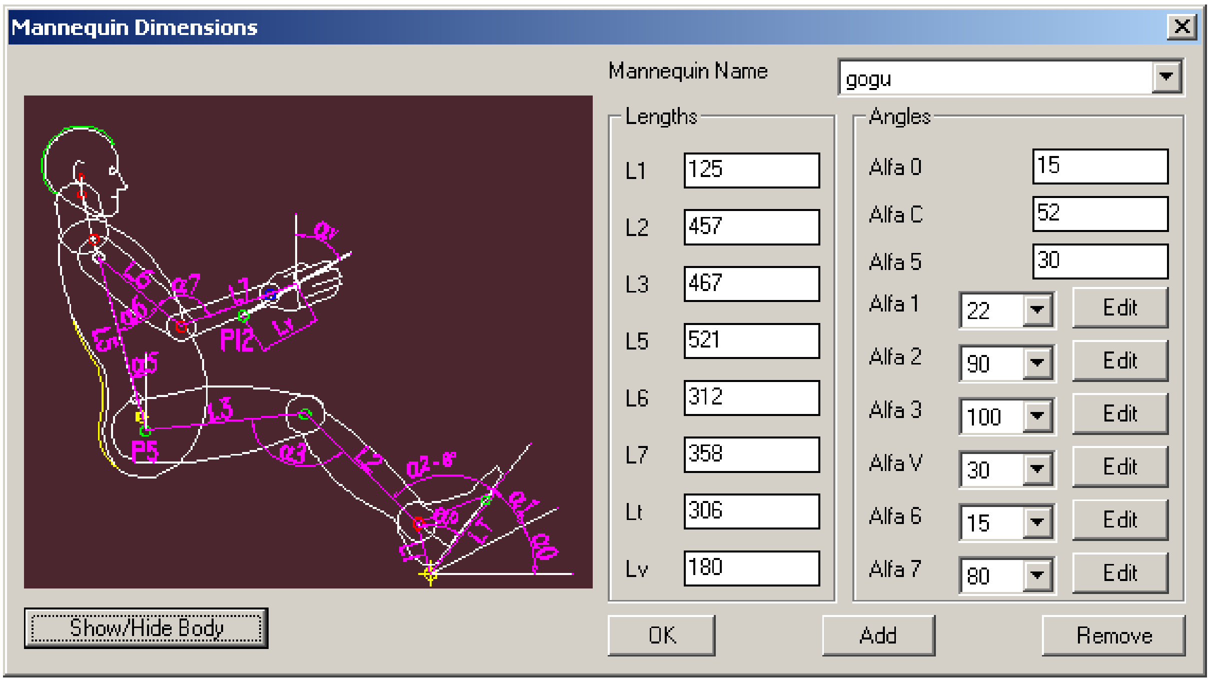
A dialog box example (for mannequin selection):



All dialog boxes are designed and managed using the DCL extension (dialog control language).

By selecting one of the names in the above dialog, all the other mannequins will be hidden. The ‘Select’ button allow to choose the active mannequin by selecting an entity (like a part of the body) on the screen.

The main dialog (see below) is a pretty complex dialog box. The preview image on the left side and the ‘Show/Hide Body’ button helps the user to identify the requested dimensions.



Mannequin Dimensions

Mannequin Name: gogu

Lengths
L1: 125
L2: 457
L3: 467
L5: 521
L6: 312
L7: 358
Lt: 306
Lv: 180

Angles
Alfa 0: 15
Alfa C: 52
Alfa 5: 30
Alfa 1: 22
Alfa 2: 90
Alfa 3: 100
Alfa V: 30
Alfa 6: 15
Alfa 7: 80

All data are saved in an external text file, with a structure similar with the standard AutoLISP lists, to easily read and transform each line in a data list. Sample code is shown below.

```
(defun read_data()      ;;read data from file
  (logger "Enter read_data")
  (setq lista_mane nil)  ;;list of all known mannequins
  (setq f (open filename "r")) ;;open the file for reading
  (if f
       (progn
            (setq s (read-line f))
            (while s
              (if (/= s "")
                   (progn
                        (setq l (read s))  ;;convert string in a list
                     (setq lista_mane (append lista_mane (list l)))
                        )) ;;if progn
              (setq s (read-line f))
              ) ;;while
            (close f)
            (setq crt_manechin_list (car lista_mane))
            (extrage_date (cdr crt_manechin_list))
                  ;;read data from the current list
            ) ;;progn
       (set_fake_data)) ;;if progn else
  (logger "Close read_data"))
```

The ‘logger’ function used in the above code is a debug function, which trace the program execution for post-running control and error trapping. The log information can be displayed on the text screen or can be saved in a text file.

CONCLUSIONS:

The new software is easy to use and also offer enough flexibility to add/modify mannequins with different dimensions, and save them in a database with “human-readable” names.

The code is well commented and modularized to allow a programmer familiarized with the AutoLISP language to maintain and improve the program; a debugging tool is also provided.

The use-case described in the first part of this paper is a typical application with real data and practical results. The necessary time for such a study is dramatically reduced and the correctness of the results is improved.