

A Virtual System for Simulation of Human Upper Limb

Valentin Grecu, Luminita Grecu, Mihai Demian and Gabriela Demian

Abstract— *The paper presents a virtual reality techniques to create and to simulate a skeletal model of human upper limbs. Motions supported by the current version of the system are: arm flexion/extension, arm adduction / abduction and forearm flexion/extension. It's present the differences between VRML and Matlab. A computer system has been developed where a set of forces done by the muscles are parameterized and the limb motion animation, in real time, is achieved through biomechanics equations.*

Index Terms — Human upper limb, modeling, virtual system, simulation GUI

1. Introduction

Motion is one of the most natural and important activities that a human being can practice, even before birth . Thus, one of the major difficulties by those who are fitted with prosthetic devices is the great mental effort needed during the first training stages.

This makes it very difficult to precisely detect the spatial position as well as the forces done by the prosthesis in replacement of the muscle. This problem is still more complicated when working with upper limb prosthesis, whose function is to simulate its biological equivalents (shoulder, arm, forearm, elbow, wrist and hand).

All these mechanisms present a very efficient system with a lot of information (degrees of freedom, speed, angles etc) to be simulated and absorbed by brain commands.

In order to help professionals dedicated to special physical needs patient rehabilitation, this work proposes a computer system which, by modeling the forces done by the muscles, is able to simulated the motions that a patient would be able to do using a upper limb prosthesis.

Most of the evaluated systems do not provide facilities which allow the musculoskeletal model be simulated, in real time, according to the forces done by the muscles. This feature is very important in the development of myoelectrical prosthesis. Besides, virtual reality techniques that support a better way of interaction and navigation with the model is also not supported.

Manuscript received March 23, 2009.

Valentin Grecu is master student with the Engineering and the Management of the Technological Systems Faculty,University of Craiova (phone: 400252205838; e-mail: valentin.grecu@cez.ro).

Luminita Grecu is lecturer PhD with the Engineering and the Management of the Technological Systems Faculty,University of Craiova (email: lumigrecu@hotmail.com).

Mihai Demian is asoc. PhD with the Engineering and the Management of the Technological Systems Faculty,University of Craiova (email: mihaidemian@yahoo.com).

Gabriela Demian is asoc. PhD with the Engineering and the Management of the Technological Systems Faculty,University of Craiova (email: gabrielademian@yahoo.com).

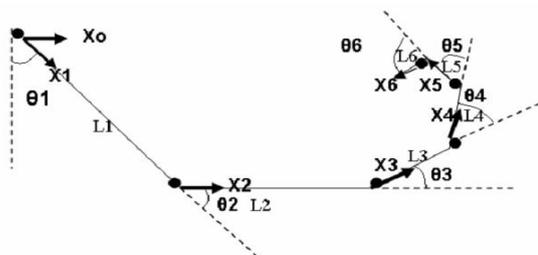


Fig. 1 Frames on each joint.

The figure 1 shows the simplified 3-dof planar human arm model with an R joint at the shoulder, elbow, and wrist pitch motions. Note this will also serve as a simplified leg model, with the hip, knee, ankle joints, and appropriate lengths and joint limits substituted for the arm parameters.

L_1 is the upper arm length, L_2 is the forearm length, and L_3 is the hand length. θ_1 is the absolute shoulder pitch angle, θ_2 is the relative elbow pitch angle, and θ_3 is the relative wrist pitch angle.

For the determination of the kinematical equations we used the Denavit-Hartenberg convention.

Write the Denavit-Hartenberg table:

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4
5	0	L_4	0	θ_5
6	0	L_5	0	θ_6
7	0	L_6	0	0

We use the following segment lengths (m):

Subject	L1 (upperarm)	L2 (forearm)	L3 (hand)
adult male	0.315	0.287	0.105
adult female	0.272	0.252	0.091

We use the following angle limits (degrees same for both male and female):

Angle	Min	Max
θ_1	-140	90
θ_2	0	145
θ_3	-70	90

We simulate Forward Pose Kinematics for the entire range of motion from the minimum to maximum angle on each joint simultaneously in MATLAB program using upper limb arm parameters (figure.2).

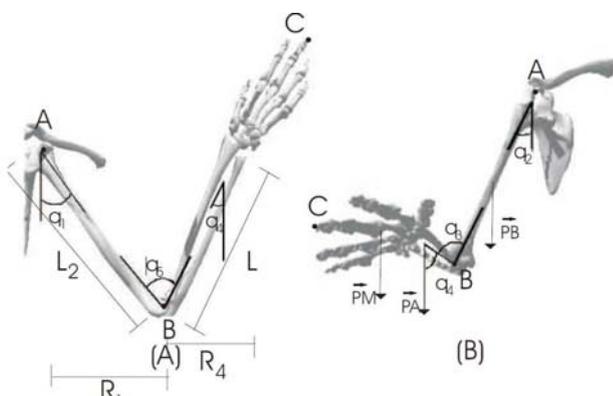


Fig. 2: Upper limb arm parameters.

The results of this human arms FPK trajectory motion simulation are given on the next section.

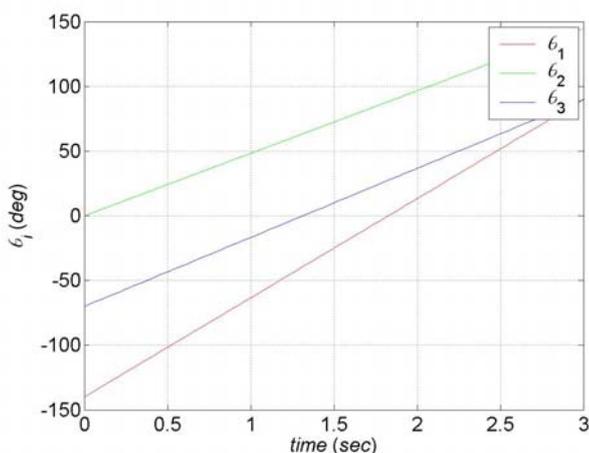


Fig.3. Input Angles

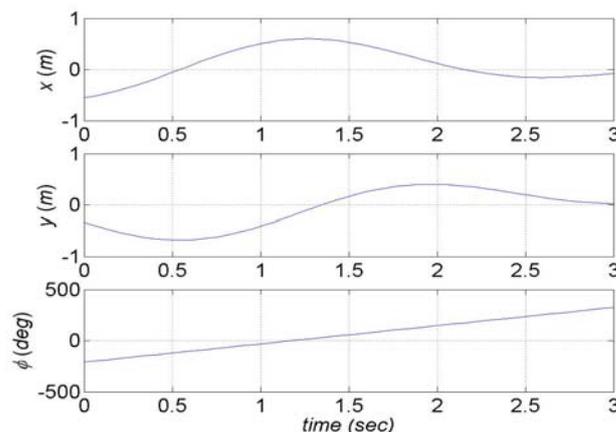


Fig.4 FPK Results (x, y, ϕ vs. t)

The Virtual Reality Toolbox is a solution for interacting with virtual reality models of dynamic systems over time. It provides a flexible MATLAB interface to virtual reality worlds. After creating MATLAB objects and associating them with a virtual world, the virtual world can be controlled by using functions and methods.

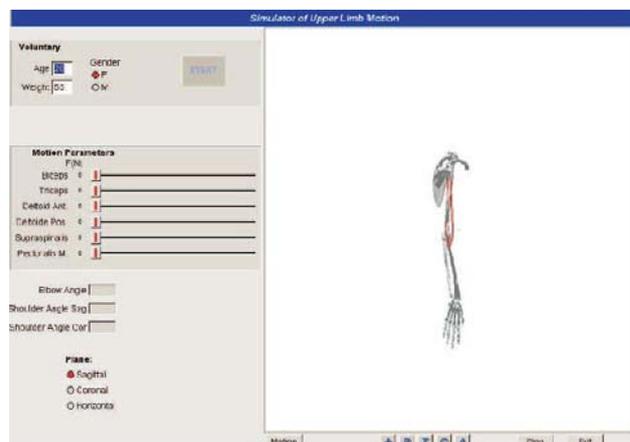


Fig.5. Human Upper Arm in a System Graphical User Interface.

From MATLAB, the user can set the positions and properties of VRML objects, create callbacks from graphical user interfaces (GUIs), and map data to virtual objects. The user can also view the world with a VRML viewer, determine its structure, and assign new values to all available nodes and their fields.

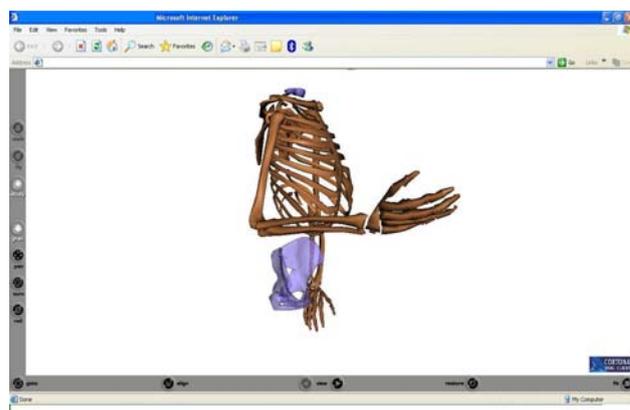


Fig.6 Human Upper Arm in a VRML Viewer.

The Virtual Reality Toolbox includes functions for retrieving and changing the virtual world properties and for saving the VRML files corresponding to the actual structure of a virtual world. MATLAB provides communication for control and manipulation of virtual reality objects using MATLAB objects.

Virtual reality headset displays with VRML could be obtained, and the VRML-file did not have to be “extended” in any way for it to be displayed within the virtual reality setup. We had access to WorldToolKit, but as the package did not fully integrate VRML .

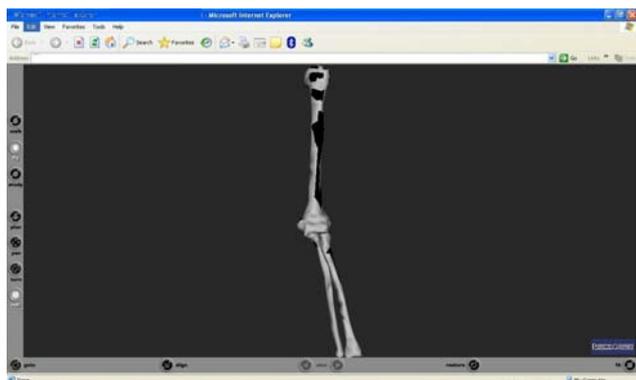


Fig.7. The bones a human upper arm in a VRML Viewer

2. VRML

We have used VRML for visualization of functional biomechanics results. Prior to Matlab 6.5 we used our own program “polyr” for isosurface generation. However, a simple iso-surface program could not suffice for the interactive visualizations we aim for.

Simple control devices (e.g., scroll bars etc) can be built, but as VRML, e.g., lacks keyboard interface complex interaction is limited, and CGI-script and java/javascript did not provide a consistent interface. Figure 7 shows a VRML visualization with a rudimentary interface, enabling hide and show of the VRML components.

Our conclusion is that VRML is a scientific visualization development environment with serious limitations, and it should serve more as a kind of Postscript/PDF for 3D interactive visualizations.

3. Matlab

Matlab is widespread in the functional bioinformatics community partly due to the program SPM.

We have had some experiences with building graphical user interfaces and simple 2D visualizations in connection with our BEAT toolbox. The Biomechanics et al. Toolbox (BEAT) Toolbox is intended to be a set of building blocks for writing analysis software.

The BEAT toolbox is not primarily a visualization environment, but rather built as a “modeler’s toolbox”, where we have provided several different analysis methods within our pluralistic analysis paradigm, and we have found Matlab to be a convenient environment for

this. We provide a simple 2D visualization for the result of the analysis. The user is able to call the function that constructs the window either from a graphical interface or from the command-line.

The size of the typical functional biomechanics data set together with Matlab’s use of double floating point numbers put a constraint on the design. The other Matlab package SPM uses memory mapping to cope with large data sets, but we chose to use only memory to give us and the user direct access to the central data matrix and ease any extensions. This will of course restrict the size of the data set one is able to work with.

Prior to version 6 there were no structures in Matlab making it difficult to represent property-like variables.

In the toolbox (which we began with Matlab 4) many global variables were necessary to handle properties.

With the structures of Matlab 5 the heterogenous data can be handle more consistently. As BrainMap data is quite heterogenous the visualization environment calls for small-scale database-like capabilities, which has to some extent been honored with Matlab 6.

From version 6.5 Matlab includes some proper 3D visualization functions, e.g., isosurface generation, and data pipes easily to VRML. We have not tried to integrate any virtual reality setup with matlab.

4. Comparison between Matlab and VRML

	Matlab	VRML
Color	indexed/RGB	RGB
Scene graph	“flat”	hierarchical
Transparency	no	yes
0-dimensional graphics	yes	(yes)
1-dimensional graphics	yes	(yes)
Volume	yes	(no)
Interactive	yes	partially
Distributable	(yes)	yes
Web-integration	(yes)	yes

Table 1: Comparison between Matlab and VRML

Matlab provides a VRML-generation function where the components of the plot/window are used to construct a VRML file. The function has not been optimal, e.g., there are no means to “plugin” to it, and text and hyperlinks are not translated.

Working with both Matlab and VRML, we find that none of them are completely optimal for all of our needs.

Table 1 sums up some differences between Matlab and VRML.

VRML uses RGB as the color model, and Matlab uses indexed and RGB color. The usual setup of a gray image for anatomical background and a hot image for a masked functional image necessitate the developer to directly control and keep track of the color scale when using indexed color. The RGB-solution is usually more convenient for the developer.

VRML is a hierarchical language where transformations (scaling, translation,...) can be nested. This approach is more developer-friendly in connection with compounded objects, such as glyphs, where components need to be

handled in a group. As Matlab is “flat” the developer will have to manipulate each component directly.

Transparency is not supported in Matlab. VRML browsers display transparent objects with varying degree of quality. In 2D plot this can be obtained by using a color scale.

In 3D without transparency one can use oblique polygons as the top-level and a wire frame at the second level. Using more than one level of wire frames it is not possible to get a clear distinctions between the different layers. Using transparent polygons more levels can be used. However, rendering quality of the transparency in VRML-browsers has not been high enough to make more than one level of transparency useful.

One and zero-dimensional graphics in the form of lines and point graphics (markers) are well supported in Matlab, and it is possible to make lines in VRML but point style, line style and thickness are not supported. Furthermore there is no zero-dimensional graphics other than a simple point.

Volumes or data structures in any higher order dimensions are not directly handled in VRML. With Matlab 6 data structures with more than two dimensions became possible.

The data is accessible on the command-line in Matlab, whereas in VRML it is hidden in the web-browser. It is very useful with an interpreted language, that does not set a shape boundary between the user and the developer.

It is desirable to have distributable file formats, e.g., the proceedings of human upper arm mapping conferences are distributed on CD-ROMs, and any visualizations could also be distributed like this. For the usefulness of the distributed file it should be in a format that a wide number is able to read and view. Standardized file formats are more likely to gain acceptance.

Matlab has a visualization distributable format: The figure/plot can be save as a “fig”-file. However, one should not pipe a fig-file to matlab directly from the web-browser, as the callback button press functions can contain malicious code.

Along with distributability comes web-integration. In VRML hyperlinks can be defined. This is also possible in Matlab as a simple function for calling the web-browser is defined. With this functionality the visualization can be made part of a larger hyperlinked web-presentation.

5. Visualization on INTERNET

The Virtual Reality Modeling Language (VRML) is a standard language to describe interactive 3-D objects and integrate them into scenes and virtual worlds. In the context of our project, VRML allows us to create interactive simulations and physical movement of the different pieces, looking for defects in the piece under analysis.

Moreover, scenes and virtual worlds can be distributed and visualized throughout Internet by means of some plug-ins developed for Web browsers. Before using VRML we needed a MATLAB-VRML converter to transfer our files to the VRML format. There is a free software called **VRMLplot**, for generating VRML files from the graphical MATLAB output (see Figure 8).

These files can be easily read by a Web browser, such as Internet Explorer, and then, the interactive visualization is

available (see Figure 7). Figure 7 shows an example of this visualization process.

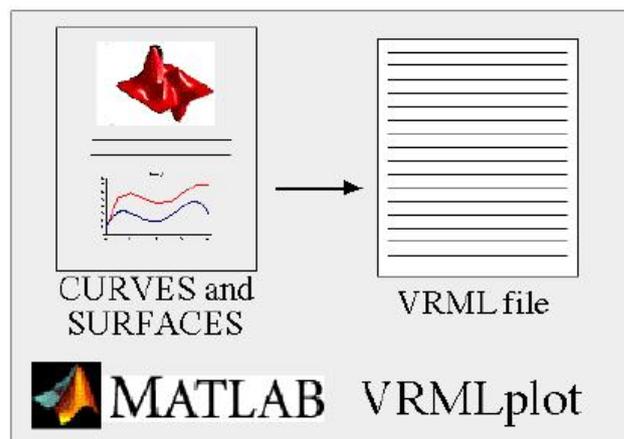


Fig. 8. Using converter MATLAB-VRML.

6. Conclusion

We find Matlab a convenient platform for development of computational algorithms, and with the visualization functions of Matlab version 6.5 a reasonable amount of visualization techniques are available.

Our experience is that most of the functionality should be accessible from a graphical interface for a toolbox to gain widespread use. At the same time it should be flexible enough for the advanced user that would like more than the “predefined interactivity”. This is supported by a general purpose interactive (programming) language.

By outputting resulting visualizations to VRML they can be distributed, e.g., on the Internet, and combined in a larger hyperlinked web-presentation.

We will pursue the combination of Matlab and VRML further for our bioinformatics toolbox.

The main contribution of this work is the simulation in real time of the joints motions of the human upper limb, obeying static equilibrium of force that are stipulated and calculated. Such biomechanics feature has not been observed in other related systems. The paper presents the mechanical analysis of human arm joints. The discussion is limited to the human shoulder, elbow and wrist. It's present the differences between VRML and Matlab. The research is then extended to the study of human manipulator. The simulation of human arm forward kinematics is performed through MATLAB Graphical User Interface and VRML.

References

- [1] Abrahams P.H., Hutchins R.T., Marks S.C. Jr., McMinns Colour Atlas of Human Anatomy, Mosby, London, 4-th ed., 1998
- [2] Abdullallah H., Tarry C., Abderrahim M, Therapeutic Robot for the Upper Limb Rehabilitation, Wseas Transaction on Systems, Issue 1, Vol. 6, 2007, pp. 88-96, 1109-2777
- [3] Netter F.H., Atlas of Human Anatomy, Second Edition, Novartis, New Jersey, 1990
- [4] Papilian V., Anatomia omului, vol. I – Aparatul locomotor, Editura ALL, Bucuresti, 1998

- [5] Panjabi M., White III, A. Biomechanics in the musculoskeletal system, Churchill Livingstone , New York, 2001
- [6] Zeid I., CAD/CAM Theory and Practice, McGraw-Hill, 1991
- [7] *** - MatLab 6.5 User Guide, The MathWorks, Inc.
- [8] Maurel, W. and Thalmann, D. (1998): Human shoulder modelling including scapulo-thoracic constraint and joint sinus cones, *Computer & Graphics*, New York, 24(2), 1998, 203-218
- [9] Lemay, M. H. and Crago, P. E. (1996), A Dinamical Model for Simulating Movements of The Elbow, Forearm, and Wrist, *Journal of Biomechanics*, 29(10), 1319-1330.
- [10] Pinho, M. S. and Santos, M. C. C. of the and KOPPER, R. A. P. (2003), Minicurso of Virtual Reality.
- [11]. B. Tondou, Estimating shoulder-complex mobility, *App. Bionics Biomech.* 4(1) (2007) 19–29.
- [12]. F .C. T. Van der Helm, Analysis of the kinematic and dynamic behaviour of the shoulder, *J. Biomech.* 27(5) (1994) 527–550.
- [13] Hall, S. J. (2000), Basic Biomechanics (Guanabara Koogan, 2000).
- [14] FAGG, A. H.(2001). The Model of Muscle Geometry is Two Degree-Of-Freedom Arm to Glide.
- [15] John J.Craig. Introduction to Robotics. 2nd ed. Pearson Education International.
- [16] V.Kumar. Introduction to Robot Geometry and Kinematics.[online]. Available from:
<http://www.seas.upenn.edu/~meam520/notes02/IntroRobotKinematics5.pdf>
- [17] Manipulator Kinematics [online]. Available from:
http://users.rsise.anu.edu.au/~chen/teaching/Robotics_ENGN4627_2005/lectureNotes/engn4627-Part03.pdf.
- [18] Nielsen, F. A and Hansen, L. K “Neuroinformatics based on VRML”. Fourth International Conference on Functional Mapping of the Human Brain. NeuroImage, ., 1998, Volume 7, pp. S782. Academic Press.
- [19] Biewener, A.A., “Biomechanics: Structures and Systems, a practical approach”, Oxford University Press, 1992
- [20] Image Acquisition Toolbox For use with MATLAB, MathWorks Inc., Natick MA, USA, 2004
- [21] Allard P., Stokes I.A.F. and Blanche J.P., “Three-Dimensional Analysis of Human Movement”, Human Kinetics, 1995