3D Visualization of Dynamical Systems for Learning Purposes

Pavol Bisták

Institute of Control and Industrial Informatics, Slovak University of Technology in Bratislava

Key words: VRML, Java, Matlab, 3D graphics

Abstract:

The paper deals with 3D visualization of dynamical system in order to better understand their behavior. For this purposes the VRML models are discussed. These models are connected with the MATLAB locally or via Internet. As an alternative approach advantages and disadvantages of 3D models created in the Blender software are mentioned.

1 Introduction

Study of dynamical systems behavior can be very difficult. Therefore it is an advantage when someone can see how these systems perform in the 3D space. Visualization in 3D not only supports imagination but also helps to understand the physical nature of processes running as a reaction to the external forces applied to these systems. Using 3D visualization the performance of different types of dynamical systems can be shown (e.g., mechanical, hydraulic, magnetic systems). Even the responses of an abstract dynamical system of the third order can be visualized in 3D space. For the visualization several software tools can be chosen. In this paper we will discuss VRML models [1] combined with Matlab and Java applets as well as the Blender program used for editing 3D graphics. After successfully completing the 3D models creation these models can play an important role in building of virtual and remote laboratories [4, 5].

2 Visualization of triple integrator using Matlab and VRML

The triple integrator dynamical system represents a theoretical problem corresponding to the simplest differential equation of the 3rd order with the nonzero right hand side. In the state space it could be described by the vector differential equation

\[ \dot{x} = Ax + bu \] (1)

\[ \begin{align*}
\dot{x} &= y \\
\dot{y} &= z ; \\
\dot{z} &= u \end{align*} \]

\[ x = \begin{bmatrix} x \\ y \\ z \end{bmatrix} ; \quad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} ; \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \]

If we suppose that the control action is constrained \( u \in \{U_1, U_2\} \) and the task is to reach the desired state in the minimum time and the additional condition is to switch between limit...
control values smoothly (exponentially decreasing changes) then this results in nonlinear
dynamic decomposition

\[ x = x_1(q_1,t_1) + x_2(q_2,t_2) + x_3(q_3,t_3) \]  
\[ x_i(q_i,t_i) = e^{-\lambda_i t_i} q_i v_i + \int_0^{t_i} e^{\lambda_i \tau} b d \tau q_i, q_i \in U_1 - \sum_{k=1}^{i-1} q_k, U_2 - \sum_{k=1}^{i+1} q_k \], \quad t_i \geq t_{i+1}, i = 1..3, t_4 = 0

where eigenvectors \( v_i \) can be computed according to chosen closed loop poles
\[ \alpha_3 < \alpha_2 < \alpha_1 < 0 \] as

\[ v_i = [\alpha_i I - A]^{-1} b \]

After solving the system on nonlinear equations (2) where parameters \( q_i \) and \( t_i \) must fulfill
limits the control law can be simply computed

\[ u = \sum_{i=1}^{3} q_i \]

This control law together with the solution of nonlinear equations is implemented in the
MATLAB Fcn block of the Simulink scheme shown in the Fig. 2.

The behavior of the triple integrator under above specified time suboptimal control is depicted
in the Fig. 1. The system moves along the trajectory towards the switching surface and then it
continues to the origin. Although the VRML model itself in this case does not provide
interactivity it is very useful for learners to express certain phases of the control process in the
phase space. Using VRML viewer learners could investigate the behavior (trajectory) from
different views and can better understand the principle of the designed control.

Fig.1: Trajectories representing the behavior of the triple integrator in the phase space built in
VRML

From the visibility reasons the individual objects in the VRML world are realized in three
dimensions, e.g., the point is representing by the ball, the trajectory by the tube. The most
complicated task has been to draw surfaces because the VRML language does not support
sophisticated mathematical expressions. Each surface has been discretized in specified points
and drawn using small elements by the connection of its vertex points.
To increase the interactivity the VRML world can be connected with the Matlab computational engine and than user can change any parameter to fully interact with the dynamical system. For instance, changing the initial conditions of integrators an arbitrary starting point can be chosen. The VR Sink block of the block diagram in the Fig. 2 assures that the changing state variables are passed to the VRML world and the user can see the movement of the ball (representing the actual state) along the trajectory.

Fig.2: The Matlab/Simulink block diagram of the triple integrator control loop connected with VRML world using VR Sink block

For engineers in order to evaluate the quality of control the responses of state variables and control value with respect to time are very important. These can be achieved using the Scope blocks of the Simulink block diagram in the Fig. 2 and are depicted in the Fig. 3

Fig.3: Responses of control value and state variables over the time

3 Visualization of real systems using Matlab, VRML and Java

Usually the VRML models are not enough interactive to fulfill all user requirements. In the previous case we avoid this by the use of the Matlab. The disadvantage of the previous solution is that the user must have Matlab installed on his computer. To cope with this problem we have designed client server application [4, 5] when it is sufficient to install the Matlab only on the server side. Clients can use the Internet connection to access Matlab services provided by the server. Then the client interface is created by the Web page and only the Web browser with the support of Java applets and JavaScript is necessary. The structure of this solution is shown in the Fig. 4.
In the Fig. 4 one can see that the VRML world is not connected directly to the server. This connection is covered by the Java applet. Then it was necessary to interconnect Java applet with VRML model so the data can be exchanged within the Web page. For this purposes we used an interlayer represented by the JavaScript code placed at the same Web page.

Besides the communication the Java applet fulfilled also another role. For instance, it was very hard to enter the numerical data into the VRML world. Instead it is possible to use sensitive nodes and slide bars as one can see in the Fig. 5. For these reasons it is also a good idea to combine a VRML world with a Java applet that supports text fields and different buttons. Thus more interactive environment can be built and in our case it was used as a user interface for virtual and remote laboratories (Fig. 5, 6).

Both user interfaces shown in the Fig. 5 and 6 consist of two main parts. The left panel serves for data entering and user interactions using different buttons and the right panel is reserved...
for visualization in 3D using VRML models. Moreover, the hydraulic system in the Fig. 5 provides additional interactivity using the slide bar on the right side and also using the sensitivity nodes for hydraulic valves that can be closed or opened by clicking the mouse.

Fig. 6: The VRML model of the helicopter rack interconnected with Java applet

4 Visualization using 3D graphics editor Blender

Blender is the open source software for creation of the 3D graphics. Originally it has not been designed for the purposes of visualization of the dynamical systems behavior. But it can be used with the Python programming language that makes it strong enough to express the responses of dynamical systems. We have tested this possibility because in comparison with VRML models Blender offers more sophisticated models with lot of predefined textures, effects, etc.

![Image of Blender 3D model]

Fig. 7: The 3D model of hydraulic system created in Blender (a) and its Control Panel created in Game Engine (b)

The disadvantage of the Blender is missing interactivity. In order to provide the interactivity features the extension with the Game Engine has to be used. In the Fig. 7a one can see the 3D model of the hydraulic system that was created in the Blender. The Fig. 7b shows the Control
Panel that provides the 3D model with the interactivity. Although this Control Panel has been created using the Game Engine, the interactivity is limited to clicking left and right mouse buttons to move slide bars and thus set up desired values.

![Virtual Laboratory room with hydraulic system](image)

**Fig. 8: The Virtual Laboratory room with hydraulic system**

The Fig. 8 represents the same hydraulic model from the previous picture but this time it is placed on the table in the virtual laboratory room and beside the model there is the Control panel. The user can walk through the room, approach different models and run virtual experiments. Up to now there are three models of dynamical systems placed in the virtual laboratory room: hydraulic system, magnetic levitation and pendulum.

## 5 Conclusion

It is well-known that modeling in the three dimensional space belongs to the time consuming activity but on the other way it can provide learners with deep insight into studied problems. Although we did not mention all problems of 3D modeling (e.g., computational load in sophisticated worlds, different players, navigation control, etc.) we can conclude that VRML model can sufficiently represent the behavior of dynamical systems for learning purposes. If we can build more precise models we need to apply more professional software tools for 3D modeling. The Blender is one possible solution that is free of charge. Its disadvantage is the limited interactivity.

Generally, we are convinced that if we create interesting 3D models with enough interactivity they will surely become a favorite tool in the learning process.

## Acknowledgements

The work has been partially supported by the Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09. It was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.
References:


Author(s):

Pavol, Bisták, Ing.
Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Institute of Control and Industrial Informatics
Ilkovicova 3, 812 19 Bratislava, Slovakia
Pavol.Bistak@stuba.sk