MMT - A WEB-BASED E-LEARNING SYSTEM FOR MATHEMATICS, MODELING AND SIMULATION USING MATLAB

Stefanie Winkler, Andreas Körner, Irene Hafner

Vienna University of Technology, Institute of Analysis and Scientific Computing
Wiedner Hauptstrae 8-10, 1040 Vienna, Austria

stefanie.winkler@tuwien.ac.at (Stefanie Winkler)

Abstract

This contribution deals with an e-learning system called MMT- Mathematics, Modeling and Tools. It is constituted to help students of Vienna University of Technology to increase their abilities on both, basic mathematics on the one hand and modeling and simulation on the other hand. This e-learning system is based on the MATLAB® Web Server Technology. Initially the MATLAB Server of version 2006 was used. As this product has been discontinued MMT has been using a self-made server containing the actual version of MATLAB. This fact supports us to update our examples for the students or create examples containing the latest tools of MATLAB, for instance SIMULINK®.

Currently there are generated examples which integrate the tools of MATLAB/SIMULINK. The system is also modified to use a Content Management System (CMS). This system supports and organizes the shareable creation and adaptation of text- and multimedia-files. The CMS will provide the means to develop and maintain the MMT examples without the need of HTML knowledge.

Keywords: e-learning system , MATLAB , SIMULINK , SIMSCAPE , CMS

Presenting Author's Biography

Stefanie Winkler. She finished the Vienna high school of music and then started studying mathematics in technology and science. She is on her way to finish the bachelor. The bachelor thesis is focused on physical modeling and simulation. A related field of work is the development of e-learning systems based on simulation environments. Additionally to the MMT she is working with Maple TA, another e-learning system.

1 General

The main target of the creation of this e-learning system are the students of Vienna University of Technology. There are many different courses, which deal with simulation and modeling. It is difficult to teach simulation and modeling without any modern facilities. MMT allows students of different fields like mathematics, geodesy, engineering, computer science and electrical engineering to use the provided examples for better understanding of modeling and simulation principles[1].

Beside the modeling and simulation part there are also examples, which help to understand mathematical basics and the influence of parameter variation. By studying the MATLAB file, which is available for each example on the MMT, the student gains insight into MATLAB and its usage. There is the possibility to download these files and edit them. The main part of the visitors of these different lessons, where the Web server application is used, have already attended courses about programming. Therefore they are familiar with reading MATLAB code.

A part of the department of analysis and scientific computing of Vienna University of Technology works a lot with simulation and modeling. Thus this institute offers many different courses about simulation in addition to the courses for mathematic basics for several studies.

In these lessons the MMT system is used to present different simulation models and explain the main aspects of modeling. This 'learning by doing' process is supported by the availability of these examples via a web interface for advanced learning at home. The relationship between these dynamical models and the reality is also a very important goal of the MMT system[2].

To assure the comprehensibility of the examples and the MATLAB codes, they underlie a certain structure. This well defined structure supports an easy implementation and illustrats special contents of teaching.

Fig.2 shows the current layout of the examples. It is separated in 3 parts. There is the header which contains the name of the example the main content the example deals with and on the right there are similar examples. The left block offers the opportunity to choose certain variables which are defined in the text above. There is also a short description of the model. The lower right part is white at the beginning. If the parameters are chosen the student confirms his choice and receives an output-image there. That was the layout used until now.

The following figure Fig.1 shows the approximately layout of the MMT e-learning system. In the section 1 the student chooses the right lecture and will get a list of all the expamles convenient to this lecture. In section 3 on the right side there will be all examples which deal with a similar topic like the chosen example. Additionally there resides the buttom 'view mfile' and the SIMULNIK block model, if the example is implemented in SIMULINK. It is also possible to upload other important files for every example. In the middle section number 2 there the description of the example, the input field and the output window are arranged.

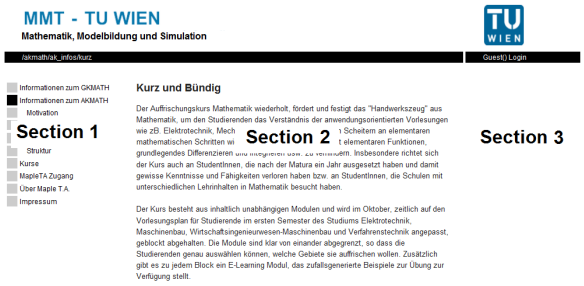

Fig. 1 This figure shows the expected layout in the CMS with the help of an other homepage of our workgroup.

In our case the interface used for input and output representation is defined by standard HTML frames. This files interact with MATLAB directly via the web server so the system becomes more stable. After defining the structur once, it is possible to adapt examples and add new ones without any code writing in PHP ot HTLM.

2 Examples with MATLAB and SIMULINK

In the following paragraph explains some examples contained in the MMT e-learning system, based on MATLAB and the MATLAB-tool SIMULINK. The advantage of SIMULINK is the clear block structure. The program code of these examples consists of many small mathematic blocks which describe every single step of the based equations. It is also possible to take a look at the based MATLAB code.

2.1 Interpolation with Splines

The first example shows an interpolation problem. It is implemented in MATLAB. The topic is the interpolation of a function which describes an increase in trapeze form by using 4 different variants.

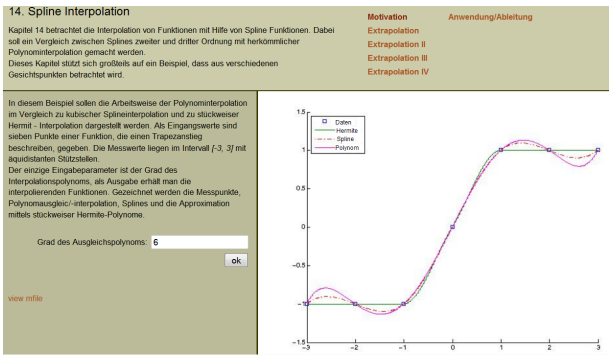

Fig. 2 An example of mathematic basics - splines

As shown in Fig.2 the different colored lines characterizes the different ways of interpolation. The green line describes the piecewise hermit-interpolation. This interpolation takes care of the derivations of the function which has to be approximated. The dashed red line

shows the interpolation with splines. Splines are defined piecewise with the additional condition that the meeting point of two polynomial pieces has to be differentiable. The used splines in this example are cubical which means that the meeting point has to be two times differentiable. The pink line describes a normal polynomial-interpolation. There is only one selectable parameter which describes the degree of the interpolation-polynomials.

In the following figure shows the influence of the chosen degree for the interpolation-polynomials.

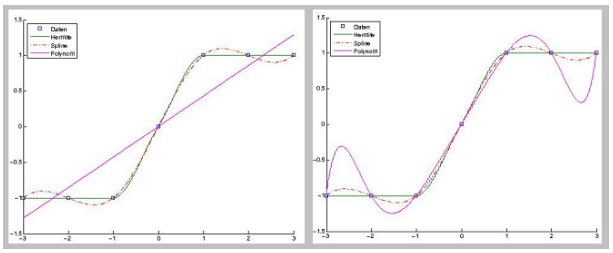

Fig. 3 Influence of different chosen degrees of the interpolation polynomial.

In Fig.2 the degree of interpolation-polynomial is 6 but in the shown Fig.3 the chosen degrees are 2 in the left one and 7 in the right one. The difference between the several interpolations depending on the chosen degree is visible.

2.2 Predator - Prey - Model

This example is implemented in MATLAB only. It describes the interaction between the population of the predators and its preys. The based equation are two first-order, non- linear differential equations, named Lotka- Volterra- equation[1]:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = x \cdot (\varepsilon_1 - \gamma_1 y) \tag{1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -y \cdot (\varepsilon_2 - \gamma_2 x) \tag{2}$$

The variable x characterizes the amount of preys and y stands for the number of predators. The parameters ε_1 describes the reproduction of the preys and γ_2 the reproduction of the predators in case of well conditions. That means that there is enough to eat and no other animal, who hunts them. The amount of preys who is killed by one predator is the constant γ_1. The last absolute term is ε_2, which describes the rate of dying predators if there is no prey. The equations eq:eqdx and eq:eqdy characterize the growth of the population of the preys and predators against the time. The fitting example in the MMT-system is shown below.

The output-image shows four different graphics. The first graphic shows the data of the predator's population. In the figure below the difference between the predator's data and the simulated population is pointed out. The upper graphic in the right shows the population of the predators and its prey. It demonstrates that there is a maximum number of preys and predators. If

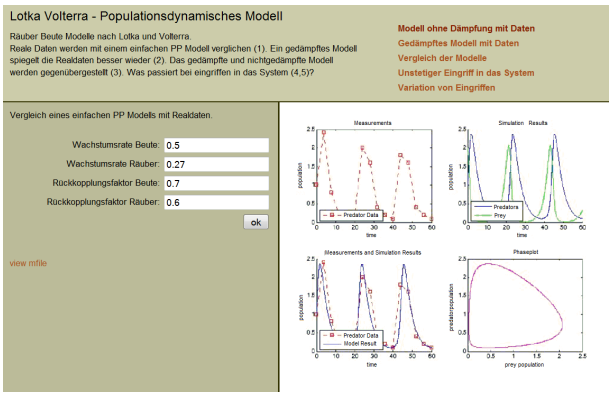

Fig. 4 This figure shows a example of simulation. The resulats shows the coherency of the population of predators and preys.

the number of predators is low the reproduction of the preys exceeds their loss and their population rises. This increase of preys leads to a rising number of the predators. There is a certain point where the killing rate of the predators is greater than the reproduction rate of the preys so the population diminishes. Because of the shrinking amount of preys there is not enough food for the predators so their count will start to fall until the starting point is reached. The last graphic lower right shows this coherency of the predators and preys population.

Among others this example is used in the course for modeling and simulation. It shows that modeling and simulation is not a abstract and academic theme but this example is useful in everyday life if someone is interested in nature and population of animals.

2.3 Pendulum

The following example [3] consists of two different SIMLULINK models, which are linked to a MATLAB main function. Both SIMULINK models, the pendulum and the free fall, are based on differential equations. The MATLAB file switches between the two models and converts the parameters for each model.

2.3.1 Description of the SIMULINK models

The pendulum model, as shown in Fig.5, is based on the differential equation

$$\ddot{\varphi} = -k\dot{\varphi} - \frac{g}{l}\sin\varphi \tag{3}$$

and describes the oscillation of a pendulum. The variables of this equation are the damping constant k, which depends on the mass, the angle φ and its derivate, the gravity force g and the length of the cord l. Except of the gravity force, which is the physical constant all these variables are chosen by the students.

Additionally there is the condition part, which controls the balance between the centripetal force and the gravity force.

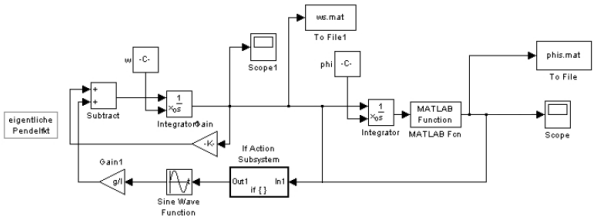

Fig. 5 The pendulum implemented in SIMULINK

$$F = -mg\cos\varphi + mL\dot{\varphi}^2 \tag{4}$$

If the force F of equation (4) gets negative the MATLAB program switches from the pendulum model to the free fall model. That means, that the gravity force outweighs the centripetal force, so the simulation of the oscillation has to be stopped and switches to the MATLAB main function. This condition is realized by the following SIMULINK model, which is integrated in the pendulum model.

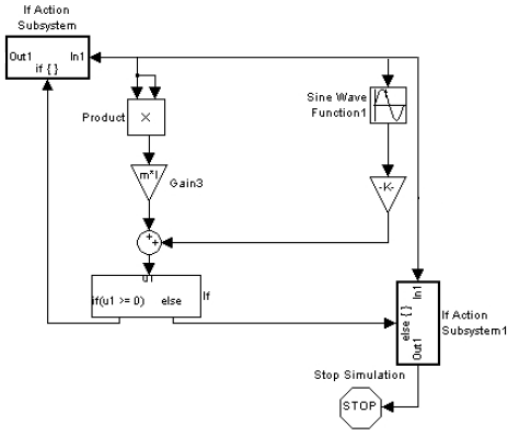

Fig. 6 The figure contains the condition request. If the condition is fulfilled the pendulum simulation goes on.

Underneath, there is the model of the free fall shown. The structure seems similar to the structure of the pendulum.

It is also based on the differential equations, which describe the x- and y-position of the mass during the free fall.

$$m\ddot{x} = -k\dot{x} \tag{5}$$

$$m\ddot{y} = -mg - k\dot{y} \tag{6}$$

These equations are also influenced by the damping constant k, the mass m and the gravity force g. Integrated in the model Fig .7 is an if-request. If the position of the mass is directly on the circular path around the origin with radius l, in other words $\sqrt{x^2+y^2} = l$ the simulation the free fall model stops.

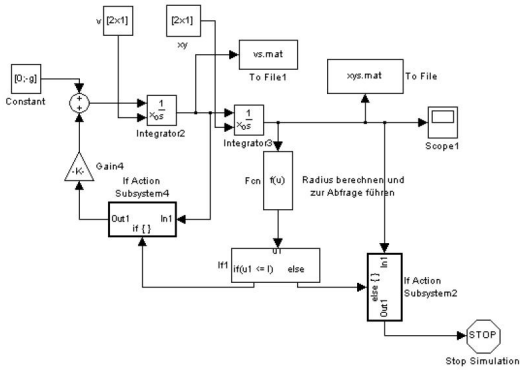

Fig. 7 The free fall model implemented in SIMULINK

2.3.2 MATLAB main function

For the pendulum model it is necessary to create a main function in MATLAB to control the sub models in SIMULINK. At first the parameters of the pendulum, which are freely chosen by the students, have to be set through an instruction in the MATLAB file.

Then the SIMULINK model starts to simulate the swinging of the mass. As described above, the simulation breaks if the gravity force outweighs the centripetal force. If the pendulum model stops the simulation the parameter-vector of the simulated values gets passed to the MATLAB file. This parameter-vector contains the angle and the angle velocity for each time step, which are used for creating the output-images. Hereafter the values have to be converted into x- and y-Value calculated from the φ-Value of the pendulum model. Below, the equations of the conversion are given.

$$(x, y) = (l\sin\varphi, l\cos\varphi) \tag{7}$$

$$(\dot{x}, \dot{y}) = (l\omega\cos\varphi, -l\omega\sin\varphi) \tag{8}$$

Subsequently the converted parameters are set in the free fall model and the main function invokes the simulation. If the coordinates fulfill the equation $\sqrt{x^2+y^2} = l$, in other words the mass reaches the circular path of the pendulum, the free fall model stops the simulation and returns the simulated values to the main file. In case of a remaining angle velocity after converting the parameters the pendulum model starts the simulation with the new values. Whenever the condition (4) gets negative the MATLAB file has to invoke the pendulum model again. This loop ends if the angle velocity value is very close to null. The transformation of the parameters from the free fall to the angle φ and the angle velocity is more complicated, as shown in the laborious If-request Fig.8 beneath.

The model creates an output-image, which shows the path the mass of the pendulum takes. Depending on the chosen parameters the resulting path variates demonstrating the influence of the students choice.

```
if x==0
    ph=pi;
else
    if y==0
        if x>0
            ph=pi/2;
        else
            ph=3*pi/2;
        end
    else
        if y>0
            if x>0
                ph=asin(x/l);          %1.Quadrant
            else
                ph=asin(y/l)+3*pi/2;   %2.Quadrant
            end
        else
            if x>0
                ph=asin((-y)/l)+pi/2;  %4.Quadrant
            else
                ph=asin((-x)/l)+pi;    %3.Quadrant
            end
        end
    end
end
```

Fig. 8 Shows the transformation of the parameters from x- and y-values to the φ-value.

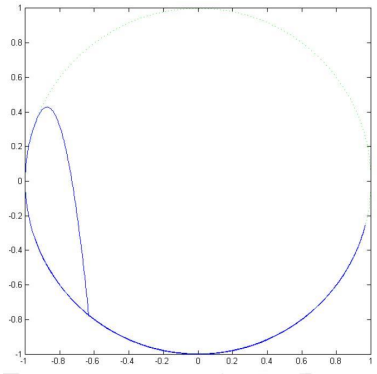

Fig. 9 The trajectory of the pendulum and free fall simulation with the parameters $k = 0.2,\ \varphi = \frac{3}{4}\pi,\ \omega = 5$

2.4 Discrete Controller

In this subsection a water level regulator is considered with draft from [4]. In modeling and simulation the control theory is an important field. This theory is easy to learn by typical examples like this water level regulator, so it is reasonable that the MMT e-learning system contains some applications in control theory.

The principle of the water level regulator is shown in Fig. 10. It consists of a water supply, a float lever and a tap. The requirement of the controller is to adjust the water inflow.

The adequat continuous controller for this requirement is a PDT_1-controller. When $U(s)$ describes the control deviation and $E(s)$ the actuating variable, for given $T_1, T_2 \in \mathbb{R}$ the equation in Laplace-region is given by

$$U(s) = \frac{T_1 s + 1}{T_2 s + 1} \cdot E(s). \tag{9}$$

In Fig. 11 the control circuit is figured. The objective is, to design a discrete controller. For this problem the discrete controller is abstract characterized in the z-domain

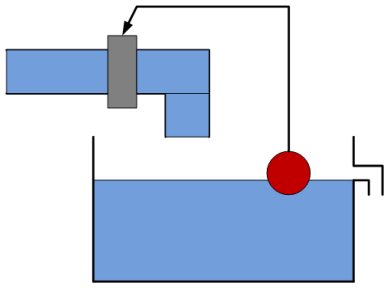

Fig. 10 Sketch of the principle of a water level regulator

as

$$U(z) = H(z) \cdot E(z) = \frac{a_0 - a_1 z^{-1}}{1 - b_1 z^{-1}} \cdot E(z), \tag{10}$$

with $a_0, a_1, b_1 \in \mathbb{R}$.

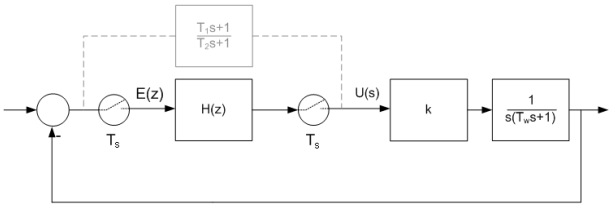

Fig. 11 Control curcuit with continuous and discrete controller

The discrete controller works with the sampling time T_s and the coefficients calculates to

$$a_0 = \frac{T_1}{T_2} e^{-T_s\left(\frac{1}{T_2} - \frac{1}{T_1}\right)}, \qquad a_1 = \frac{T_1}{T_2} e^{-\frac{T_s}{T_2}}, \tag{11}$$
$$b_1 = e^{-\frac{T_s}{T_2}},$$

where T_1 and T_2 are the delay time and the derivative time of the continuous controller. This both control strategies, the continuous and the discrete controller is implemented and compared in SIMULINK, see Fig. 12.

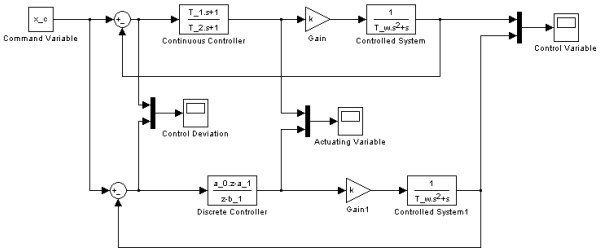

Fig. 12 Control curcuit with continuous and discrete controller implemented in SIMULINK

The scopes of the model in Fig.12, allow the students to view the results of the simulation. They can observe the

influence of the sampling time T_s on the performance of the discrete controller. Another aspect is the real technical implementation of the discrete controller. In technical systems they are realized via a micro controller and it is possible that a failure occurs. This time lag effects a failure in the behavior of the controller. For the observation of this influence a MATLAB-function is provided on MMT e-learning system. The system differentiates between a failure, when the micro controller stores the last valid actuating variable longer than scheduled or the micro controller produces a zero signal at the output. In Fig. 13 the case of storing the last valid actuating variable is shown.

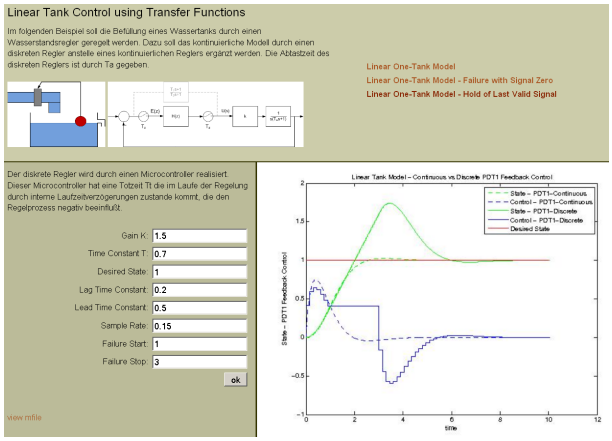

Fig. 13 Discrete controller with failure

3 Future Improvements

3.1 CMS - Content Management System

Another development, beside new and revised examples, is the modification of MMT to use a Content Management System (CMS). This system supports and organizes the shareable creation and adaptation of text- and multimedia-files. The CMS will provide the means to develop and maintain the MMT examples without the need of HTML knowledge. This CMS is already used in combination with another e-learning system. The difference is that, there the CMS is only used for preparation of information and administration.

In this context the CMS is used to integrate the examples directly on the website without a link and the use of a password to open the link. In CMS it is possible to connect to another system, in case of Technical University of Vienna the TUWEL system (TU Vienna e-learning system). If the student attends a certain lecture the system shows automatically all subscribed courses and the adapted examples. So the CMS facilitates the handling of MMT for students as well as for the administrators.

Because of the modification of MMT there is a new layout of the examples which is shown in Fig.1 and described there below.

3.2 MATLAB tools

Currently there are generated examples which integrate MATLAB and the tool of MATLAB/SIMULINK. There are very different MATLAB examples, easier ones for the mathematic basic lectures and other ones for courses about modeling and simulation. The student can study the MATLAB code-file which contains the main instructions for building such models. In addition the student has the opportunity to download the MATLAB file to edit and to test this file and learn more about programming. All these programs have to be suitable and well-structured program files to relieve comprehensibility.

In contrast SIMULINK provides a set of blocks presenting data in- and output, mathematical and logical operations. Connectors stands for the flow of data between the separate blocks. In SIMULINK equations can be implemented combining these blocks. To look ahead there will be more examples generated containing SIMULINK models.

Another innovation will be the embedding of examples working with other MATLAB tools, for instance SIMSCAPE. SIMSCAPE is another block-structured simulation tool of MATLAB. This tool simplifies representing physical structures because it is not necessary to formulate the based mathematical equations. It has also a block library and the examples arevdeveloped by combining these blocks. But the blocks differ from the blocks of the SIMULINK library. Below there is shown a model in SIMSCAPE which describes the movement of a pendulum without a free fall. Comparing Fig.5 and Fig.14 the difference between SIMULINK and SIMSCAPE becomes clear.[5]

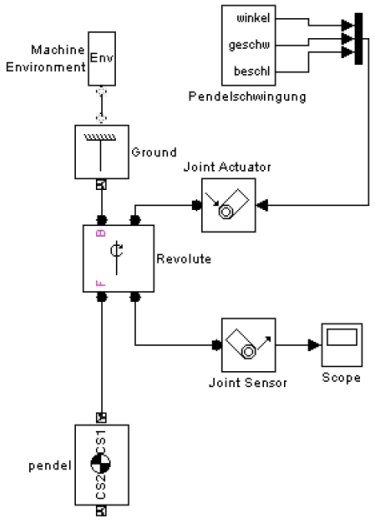

Fig. 14 The bar pendulum modeled with SIMSCAPE.

The model consists of much less blocks and the blocks differ from the SIMULINK ones. There is a ground connected to a revolute-block and the body. The axis of rotation is set within the revolute-block. In the body-block the coordinates of the position are set. The, so

called, joint actuator block is necessary for the damping, in other words the revolution is influenced by the incoming differential equation (3) of the pendulum. In SIMCAPE there is also the possibility to take a look at a three-dimensional animation of the model, which is very useful to spark the students interest in simulation.

4 Conclusion

The main aspect of the MMT-system is to give the students the possibility to get a connection to the modeling and simulation. On the one hand there are many examples which help the students to test different parameters and their impact on the other hand there are shown different ways to implement examples of modeling and simulation, like MATLAB and SIMULINK. They get to know a lot of connected fields and applications of modeling like control theory and simulation as shown in the pendulum example. The pendulum especially also shows whow to connect to different models, which is also a very important part of simulation.

Additionally there is the background part. The conversion to CMS helps the lecturer and his assistants to coordinate the examples to the right lectures. There is also the permission point which is controlled by the web system as well.

5 References

[1] G. Schneckenreither A. Körner, G. Zauner. *Ein e-learning System fr MMT - Mathematik, Modellbildung und Tools, Systemerweiterung und Einbindung von graphischer Modellbildung*. Institut of analysis and scientific computing and DWH Simulation Services, Proceeding: ASIM conference in Cottbus, 2009.

[2] F. Breitenecker G. Zauner, N. Popper. *A php/MATLAB based e-learning system for education in engeineering mathematics and in modeling and simulation*. Institut of analysis and scientific computing, Proceeding: 6th Eurosim Ljubljana, 2007.

[3] S. Winkler. *Bachelor thesis: Modellierung eines Fadenpendels mit Überschlag in Simulink*. Institut of analysis and scientific computing, Vienna, 2009.

[4] F. Breitenecker, H. Ecker, and I. Bausch-Gall. *Fortschritte in der Simulationstechnik Band 2: Simulation mit ACSL*. Vieweg, Braunschweig, 1993.

[5] I. Hafner. *Bachelor thesis: Modellierung eines gedämpften Stabpendels in Simulink mit mathematischer Aufbereitung*. Institut of analysis and scientific computing, Vienna, 2009.